Compiladores

Implementando un Parser LR

Contrucción de parser engines
SLR(1), LR(1) y LALR(1)

Resumen

- Resumen de la construcción de un parser LR(0)
- Limitaciones de los lenguajes LR(0)
- Construcción de un parser engine SLR(1)
- Limitaciones de los lenguajes SLR(1)
- Construcción de un parser engine LR(1)
- Construcción de un parser engine LALR(1)

Items LR(0)

- Tenemos que capturar cuánto de una producción hemos escaneado hasta ahora

$$\mathbf{<X>} \rightarrow (\quad \mathbf{<X>} \quad)$$

- Representado por 4 ítems
  - $\mathbf{<X>} \rightarrow \bullet (\quad \mathbf{<X>} \quad)$
  - $\mathbf{<X>} \rightarrow (\bullet \mathbf{<X>} \quad)$
  - $\mathbf{<X>} \rightarrow (\quad \mathbf{<X>} \bullet)$
  - $\mathbf{<X>} \rightarrow (\quad \mathbf{<X>} \quad) \bullet$

Ejemplo de ítems

- La gramática
  - **<S> → <X> $**
  - **<X> → (<X>)**
  - **<X> → ( )**
- Items
  - **<S> → • <X> $**
  - **<S> → <X> • $**
  - **<X> → • (<X>)**
  - **<X> → (• <X>)**
  - **<X> → (<X> •)**
  - **<X> → (<X>) •**
  - **<X> → • ( )**
  - **<X> → (•)**
  - **<X> → ( ) •**

Idea clave de los Items

- Si el "estado actual" contiene el ítem $A \rightarrow \alpha \bullet c\ \beta$ y el símbolo actual en el buffer de entrada es c
  - El estado le dice al parser que ejecute un shift
  - El siguiente estado va a contener $A \rightarrow \alpha\ c \bullet \beta$
- Si el "estado" contiene el ítem $\mathbf{A \rightarrow \alpha \bullet}$
  - El estado le dice al parser que ejecute un reduce
- Si el "estado" contiene el ítem $S \rightarrow \boldsymbol{\alpha} \bullet \$$ y el buffer de entrada está vacío
  - El estado le dice al parser que ejecute un accept

Closure() de un conjunto de ítems

- Closure encuentra todos los ítems en el mismo "estado"
- Algoritmo para **closure(I)**
  - Todo ítem en **I** es también un ítem en **closure(I)**
  - Si $A \rightarrow \alpha \bullet B \beta$ está en **closure(I)** y $B \rightarrow \bullet \gamma$ es un ítem, entonces agregamos $B \rightarrow \bullet \gamma$ a **closure(I)**
  - Repetir hasta que no se puedan agregar más ítems a **closure(I)**

Ejemplo de Closure

- Encontrar **closure**$(\langle X\rangle \rightarrow (\bullet \langle X\rangle))$

$$\left\{ \begin{array}{l} \langle X\rangle \rightarrow (\bullet \langle X\rangle) \\ \langle X\rangle \rightarrow \bullet (\langle X\rangle) \\ \langle X\rangle \rightarrow \bullet ( ) \end{array} \right\}$$

- Items

$\langle S\rangle \rightarrow \bullet \langle X\rangle \$$

$\langle S\rangle \rightarrow \langle X\rangle \bullet \$$

$\langle X\rangle \rightarrow \bullet (\langle X\rangle)$

$\langle X\rangle \rightarrow (\bullet \langle X\rangle)$

$\langle X\rangle \rightarrow (\langle X\rangle \bullet)$

$\langle X\rangle \rightarrow (\langle X\rangle) \bullet$

$\langle X\rangle \rightarrow \bullet ( )$

$\langle X\rangle \rightarrow (\bullet)$

$\langle X\rangle \rightarrow ( ) \bullet$

Goto() de un conjunto de ítems

- Goto encuentra el nuevo estado después de consumir un símbolo de la gramática mientras estamos en el estado actual

- Algoritmo para **goto(I, X)**
  donde **I** es un conjunto de ítems
  y **X** es un símbolo de la gramática

$$\textbf{goto(I, X) = closure(} \left\{ A \rightarrow \alpha X \bullet \beta \;\middle|\; A \rightarrow \alpha \bullet X \beta \text{ en } I \right\})$$

- Goto es el nuevo conjunto obtenido al "mover el punto" sobre X

Ejemplo de Goto

- Encontrar **goto(**$\langle X\rangle \rightarrow (\bullet \langle X\rangle)$, $\langle X\rangle$ **)**

$$\left\{ \langle X\rangle \rightarrow (\quad \langle X\rangle \quad \bullet \quad) \right\}$$

Items

$\langle S\rangle \rightarrow \bullet \langle X\rangle \ \$$

$\langle S\rangle \rightarrow \langle X\rangle \bullet \$$

$\langle X\rangle \rightarrow \bullet (\quad \langle X\rangle \quad)$

$\langle X\rangle \rightarrow (\bullet \langle X\rangle \quad)$

$\langle X\rangle \rightarrow (\quad \langle X\rangle \bullet)$

$\langle X\rangle \rightarrow (\quad \langle X\rangle \quad) \bullet$

$\langle X\rangle \rightarrow \bullet (\ )$

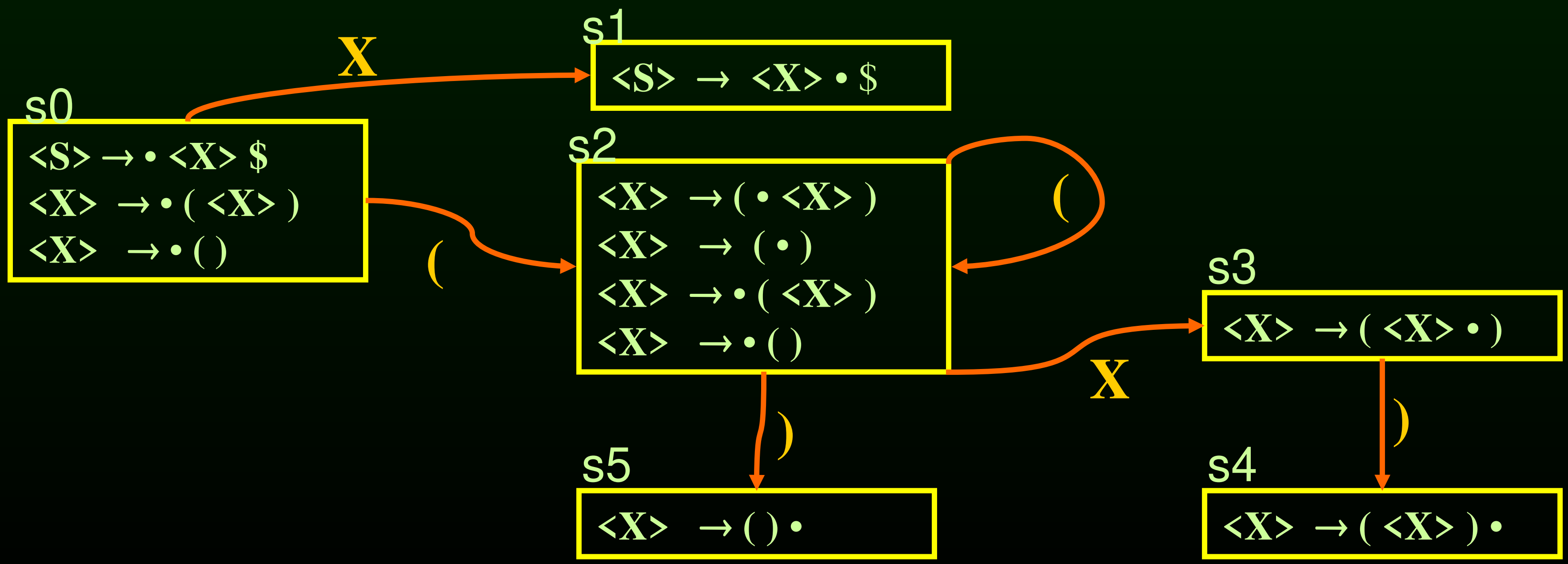
$\langle X\rangle \rightarrow (\bullet)$

$\langle X\rangle \rightarrow (\quad) \bullet$

Construyendo los estados del DFA

- Comenzamos con la producción **<S'>** $\rightarrow$ **• <S>** $
- El primer estado es **closure(**<S'> $\rightarrow$ • <S> $**)**
- Elegimos un estado **I**
  - Para cada A$\rightarrow$ α • X β en **I**
    - encontrar **goto(I, X)**
    - si **goto(I, X)** no es ya un estado, creamos uno
    - Agregamos una arista X del estado **I** al estado **goto(I, X)**
- Repetimos hasta que no sea posible agregar nada más

Ejemplo de construcción de los estados del DFA

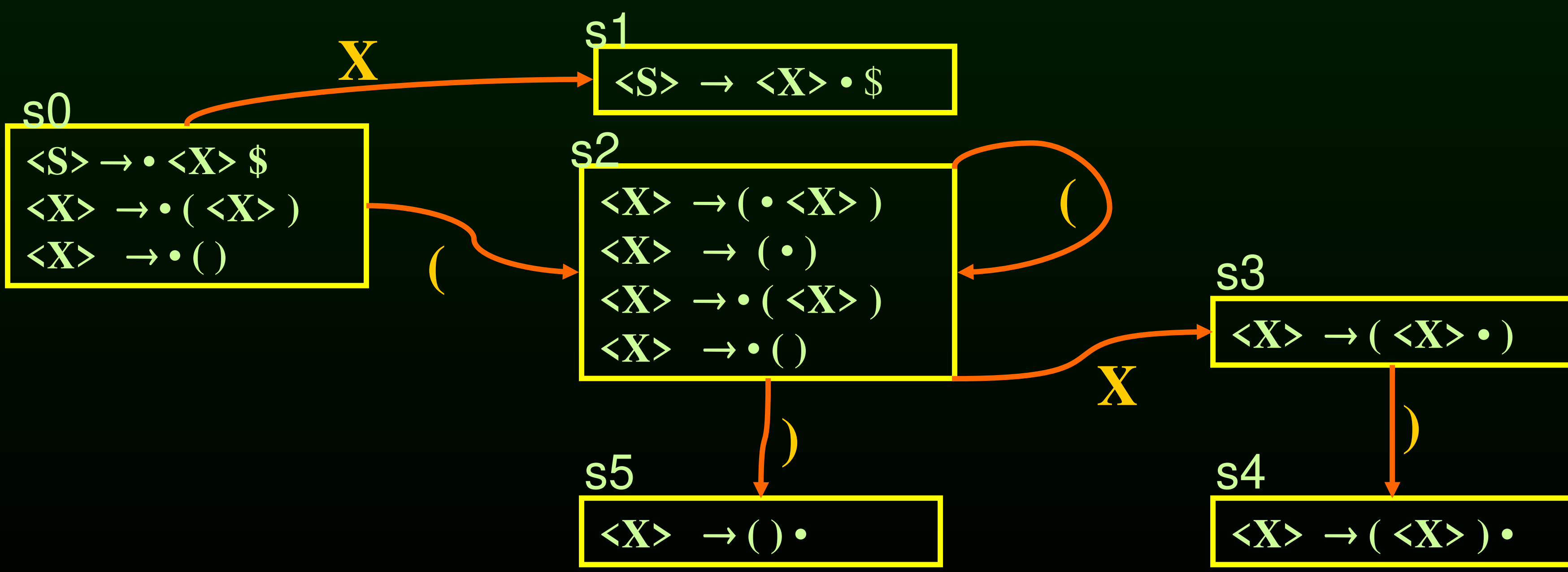


Creando las tablas de parseo

- Para cada estado
  - Transición a otro estado usando un símbolo terminal es un shift a ese estado (*shift to sn*)
  - Transición a otro estado usando un no-terminal es un goto a ese estado (*goto sn*)
  - Si hay un ítem $\mathbf{A} \rightarrow \alpha \bullet$ en el estado hacemos una reducción con esa producción para todos los terminales (*reduce k*)

Ejemplo de Construcción de Parse Table

State	ACTION (	)	$	Goto X
s0	shift to s2	error	error	goto s1
s1	error	error	accept	
s2	shift to s2	shift to s5	error	goto s3
s3	error	shift to s4	error	
s4	reduce (2)	reduce (2)	reduce (2)	
s5	reduce (3)	reduce (3)	reduce (3)	



Construcción de un Parse Engine LR(0)

- Agregamos la producción especial **S’** → **S** $
- Encontramos los ítems de la CFG
- Creamos el DFA
  - Usando las funciones **closure** y **goto**
- Construimos la tabla de parseo

Resumen

- Resumen de la construcción de un parser LR(0)
- Limitaciones de los lenguajes LR(0)
- Construcción de un parser engine SLR(1)
- Limitaciones de los lenguajes SLR(1)
- Construcción de un parser engine LR(1)
- Construcción de un parser engine LALR(1)

Ejemplo

- Strings de uno o más paréntesis izquierdos seguidos del mismo número de paréntesis derechos

  <S> → <X> $

  <X> → (<X>)

  <X> → ( )

- Strings de cero o más paréntesis izquierdos seguidos del mismo número de paréntesis derechos

Ejemplo

- Strings de uno o más paréntesis izquierdos seguidos del mismo número de paréntesis derechos

  <S> → <X> $

  <X> → (<X>)

  <X> → ( )

- Strings de cero o más paréntesis izquierdos seguidos del mismo número de paréntesis derechos

  <S> → <X> $

  <X> → (<X>)

  <X> → ε

Ejemplo

- La gramática

  <S> → <X> $

  <X> → (<X>)

  <X> → ε

- Items

  <S> → • <X> $

  <S> → <X> • $

  <X> → • (<X>)

  <X> → (• <X>)

  <X> → (<X> •)

  <X> → (<X>) •

  <X> → ????

Ejemplo

- La gramática

  <S> → <X> $

  <X> → (<X>)

  <X> → ε

- Items

  <S> → • <X> $

  <S> → <X> • $

  <X> → • (<X>)

  <X> → (• <X>)

  <X> → (<X> •)

  <X> → (<X>) •

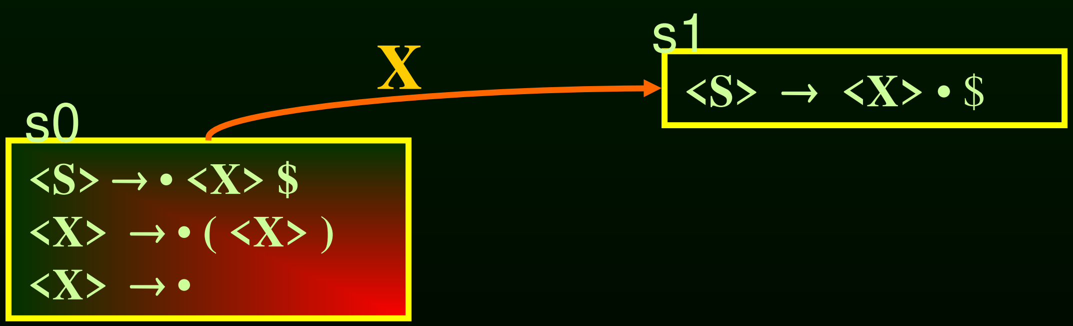
  <X> → •

Construcción del DFA para el Ejemplo

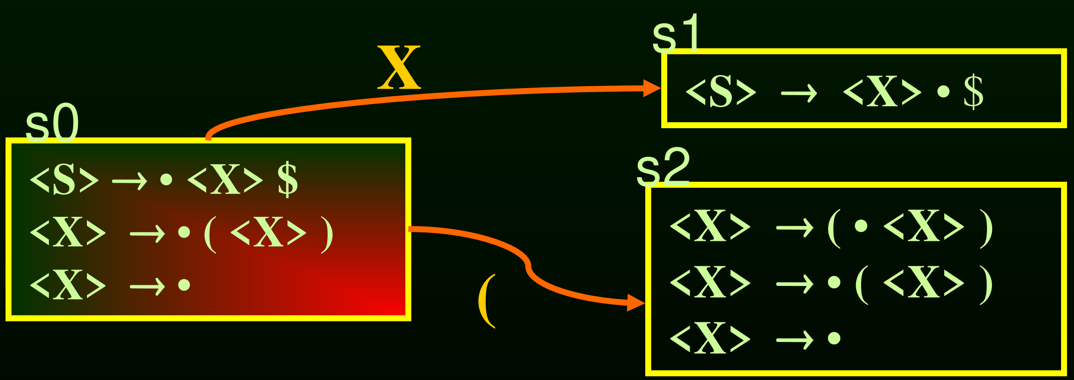
s0

<S> → • <X> $
<X> → • (<X>)
<X> → •

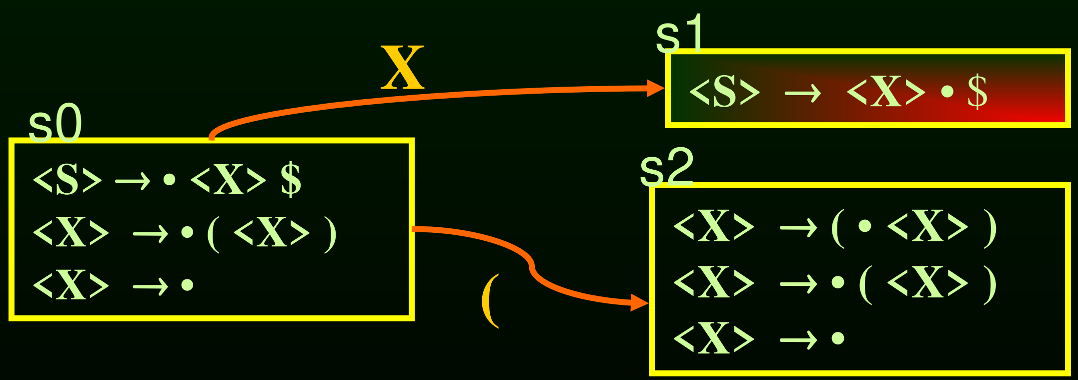
Construcción del DFA para el Ejemplo



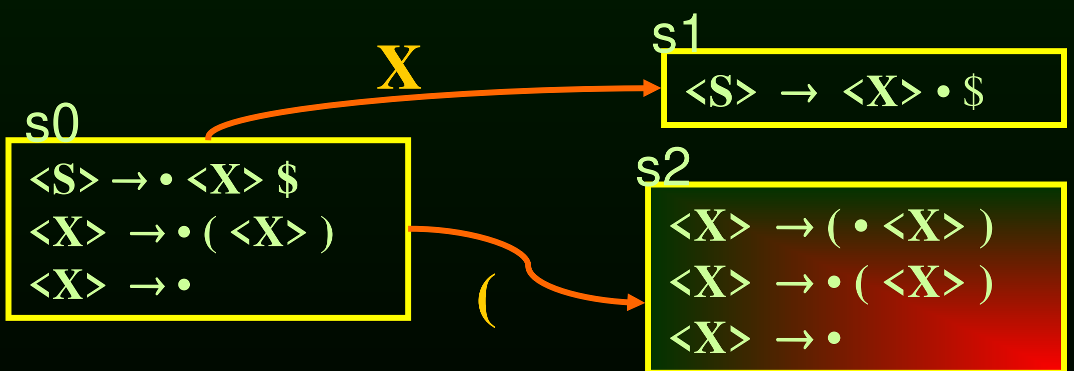
Construcción del DFA para el Ejemplo



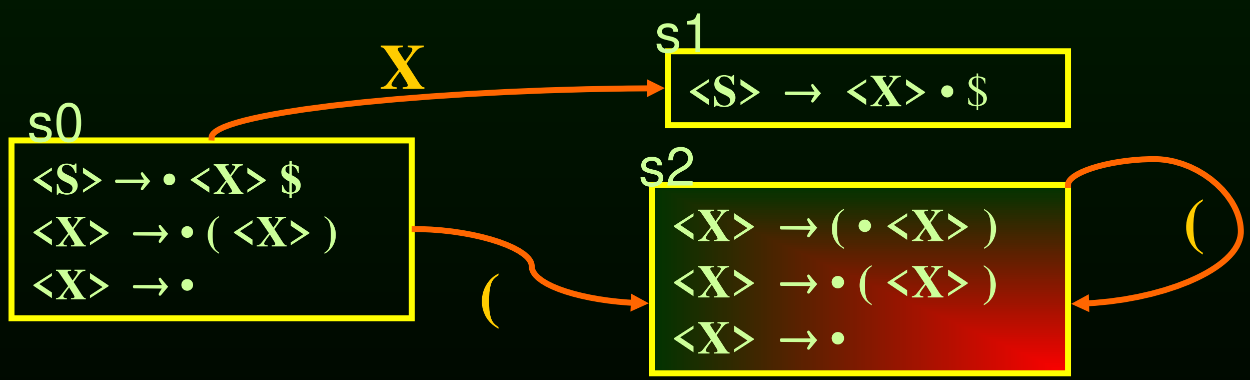
Construcción del DFA para el Ejemplo



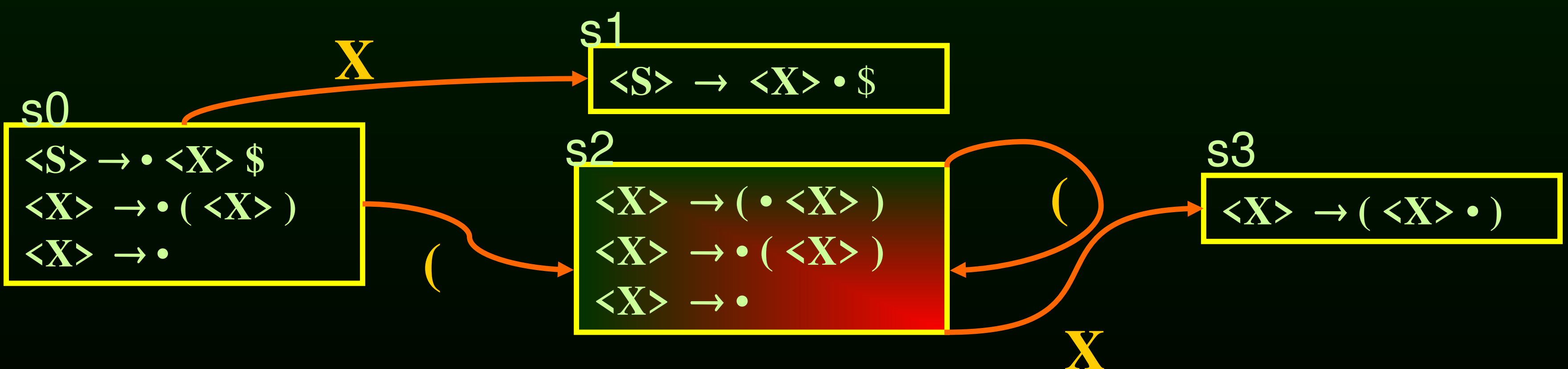
Construcción del DFA para el Ejemplo



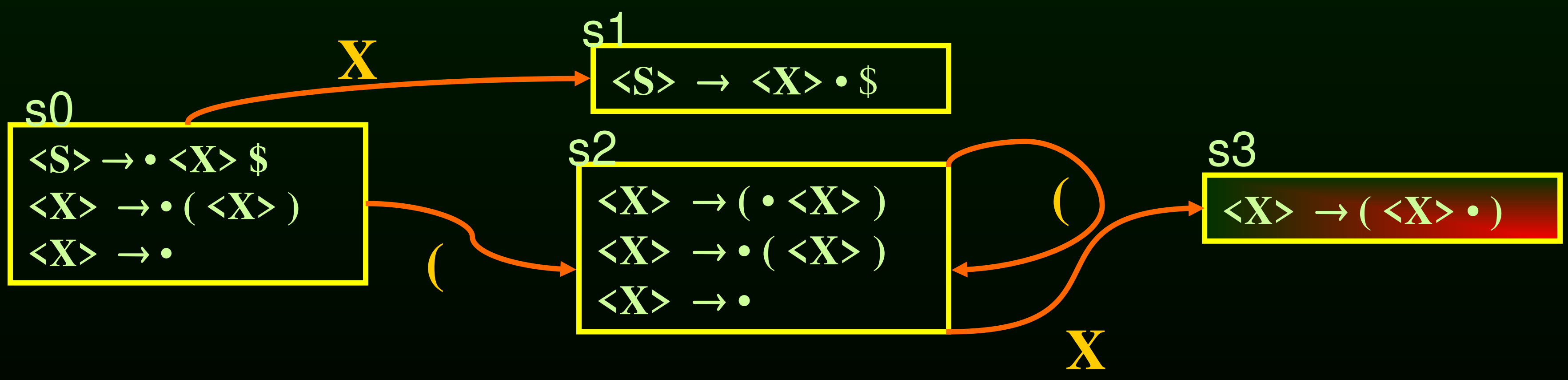
Construcción del DFA para el Ejemplo



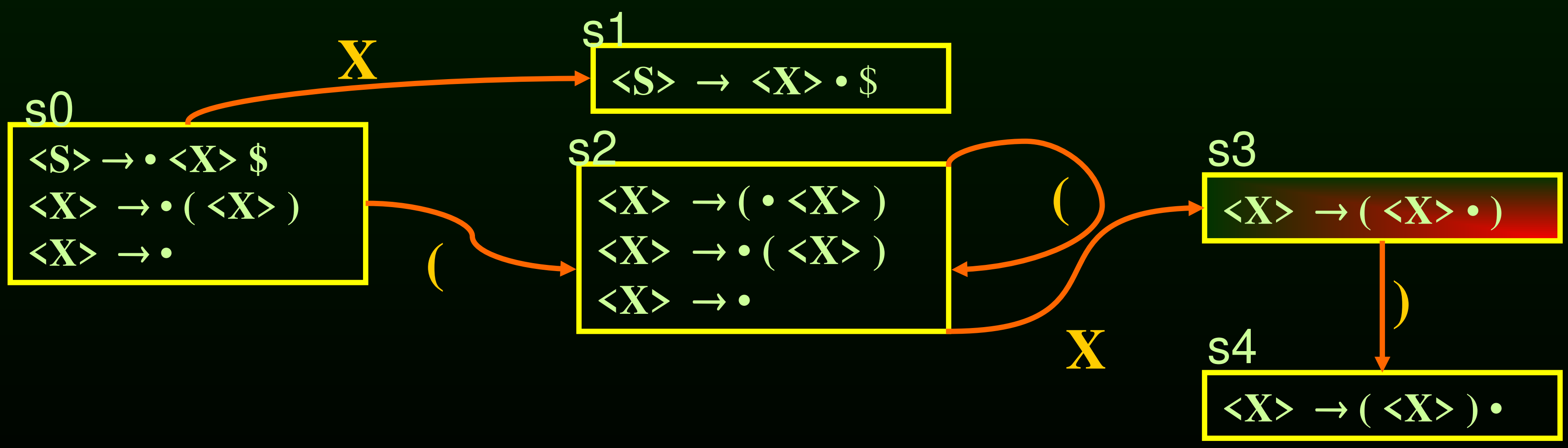
Construcción del DFA para el Ejemplo



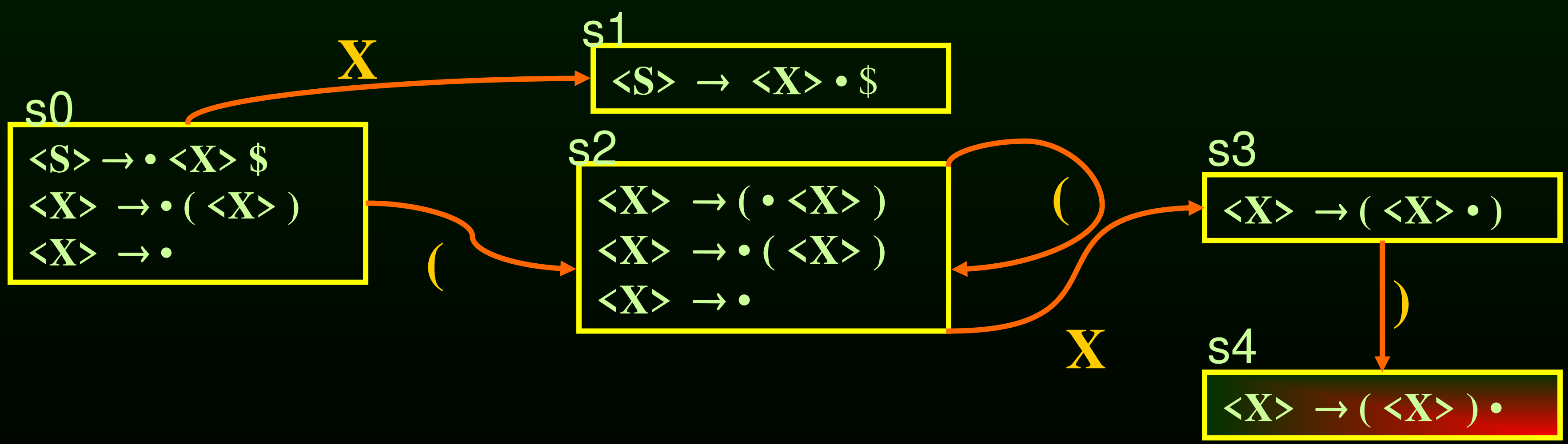
Construcción del DFA para el Ejemplo



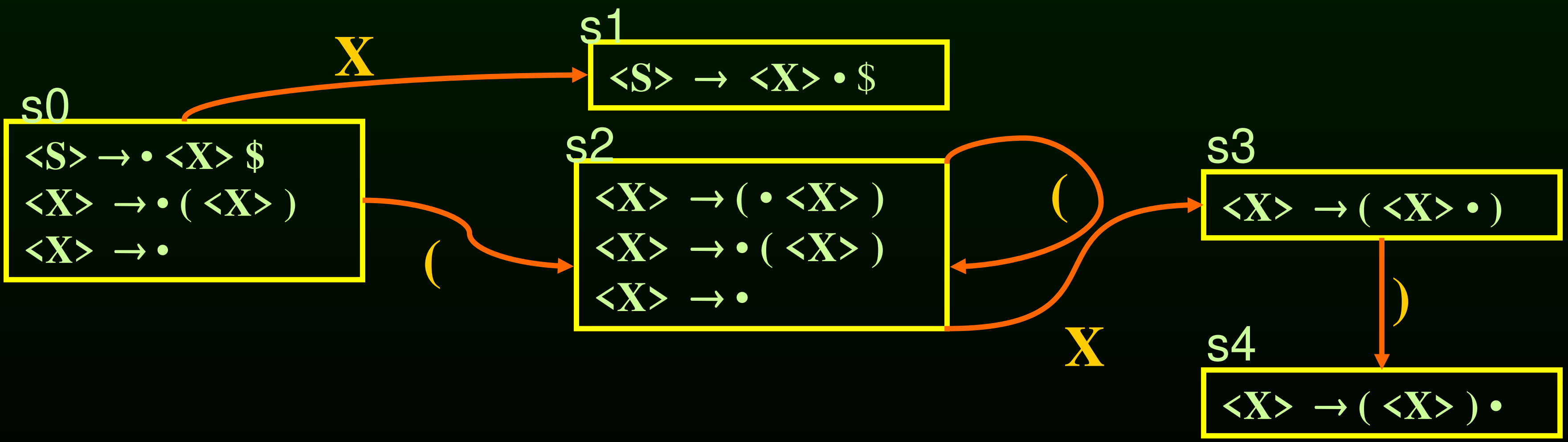
Construcción del DFA para el Ejemplo



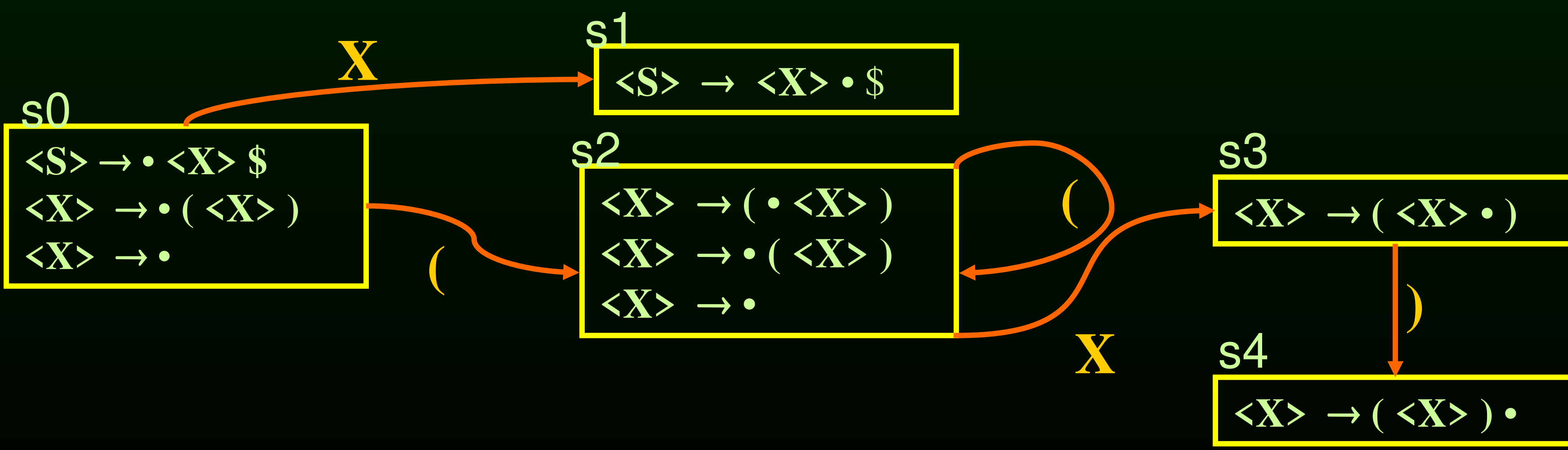
Construcción del DFA para el Ejemplo



Construcción del DFA para el Ejemplo

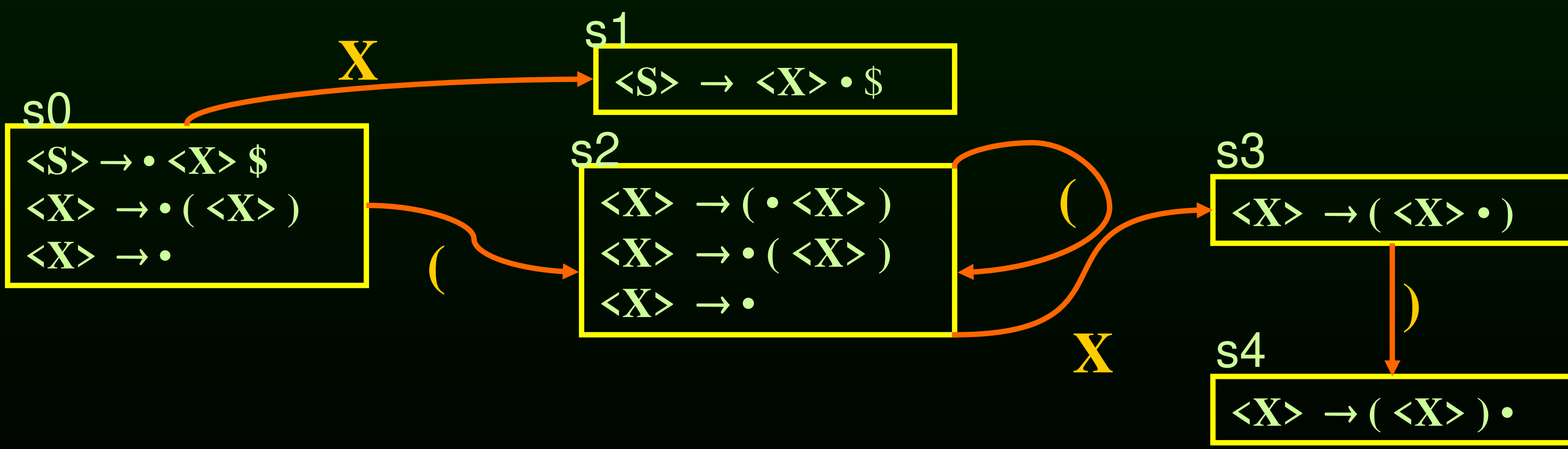


Construcción de la tabla de parseo para el Ejemplo



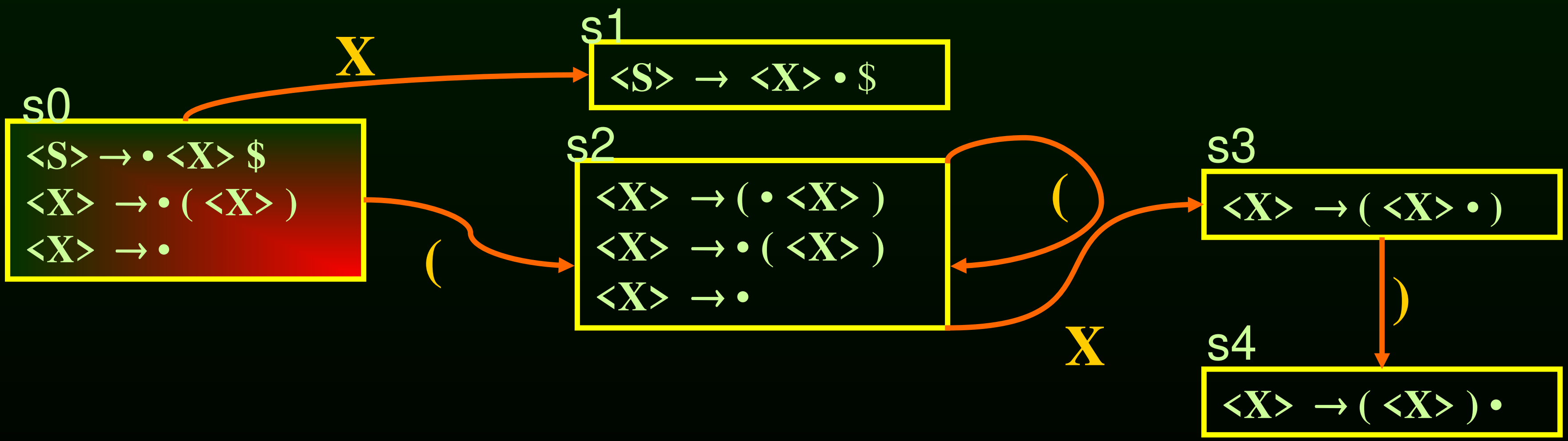
Construcción de la tabla de parseo

	ACTION			Goto
State	(	)	$	X
s0				
s1				
s2				
s3				
s4				



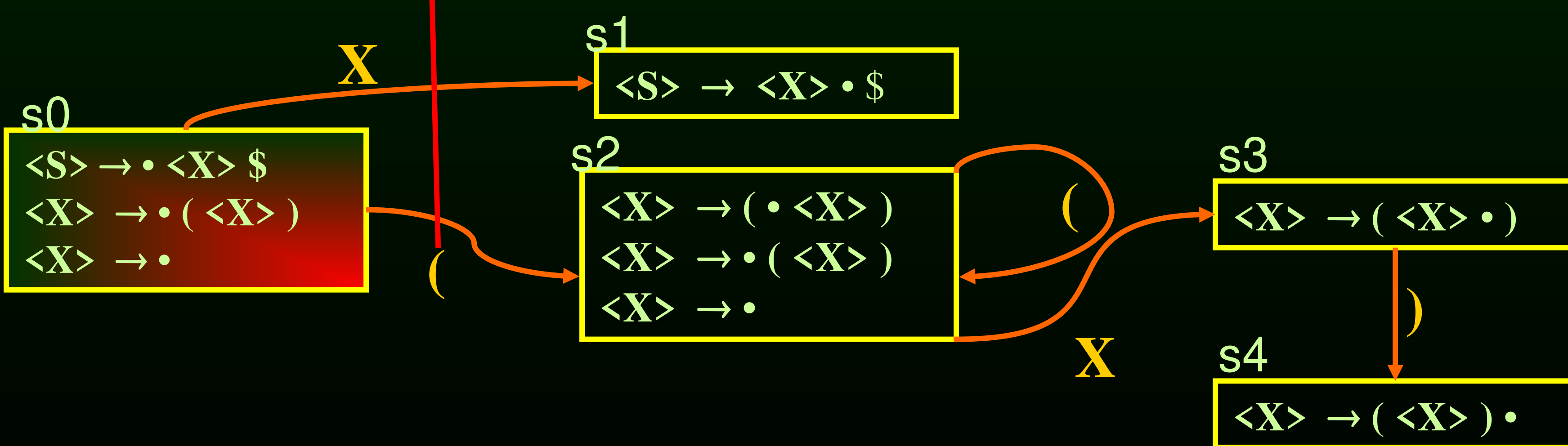
Construcción de la tabla de parseo

	ACTION			Goto
State	**(**	**)**	**$**	**X**
s0				
s1				
s2				
s3				
s4				



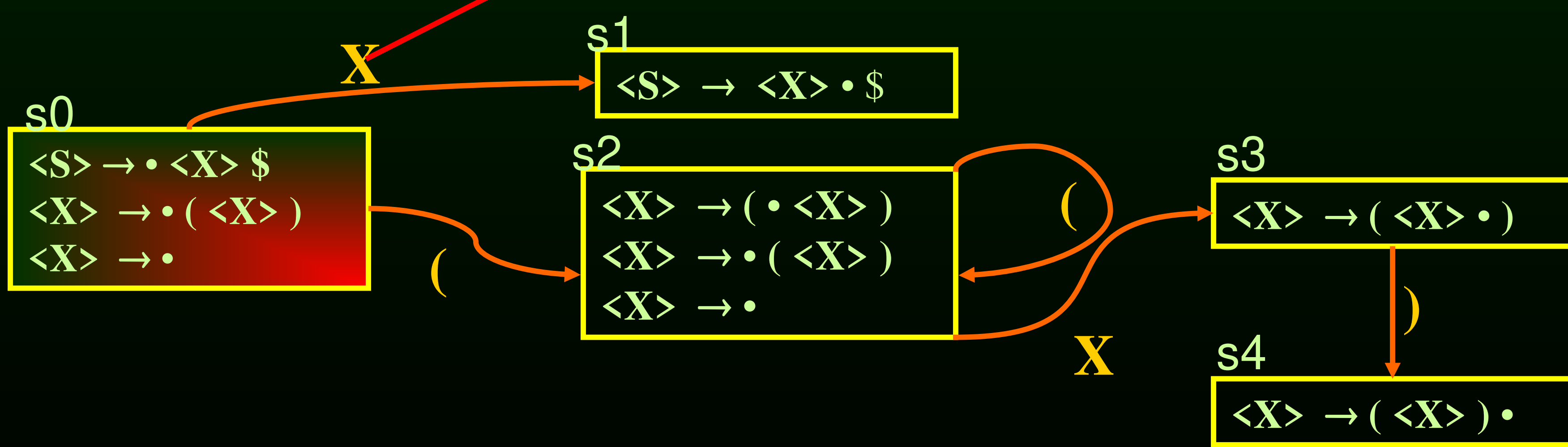
Construcción de la tabla de parseo

	ACTION			Goto
State	(	)	$	X
s0	shift to s2			
s1				
s2				
s3				
s4				



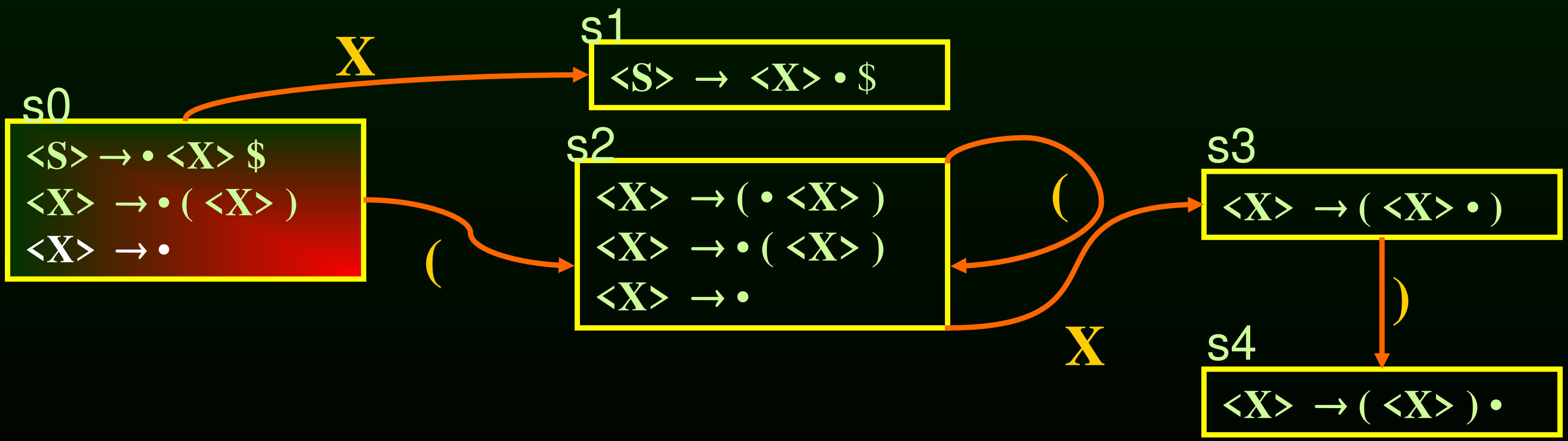
Construcción de la tabla de parseo

		ACTION		Goto
State	**(**	**)**	**$**	**X**
s0	shift to s2			goto s1
s1				
s2				
s3				
s4				



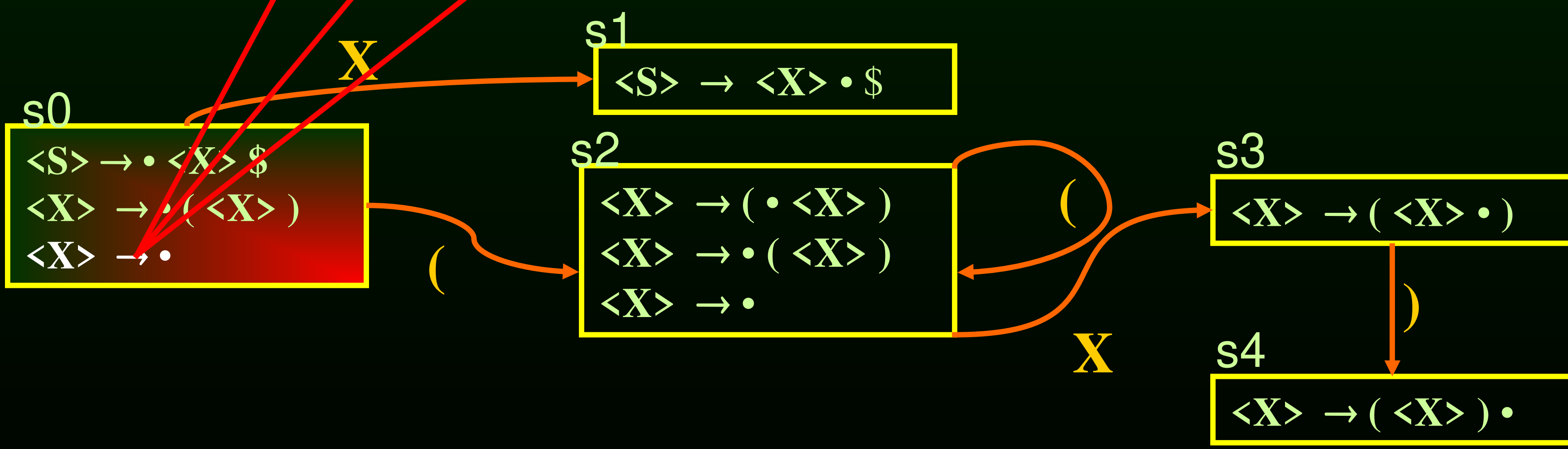
Construcción de la tabla de parseo

	ACTION			Goto
State	(	)	$	X
s0	shift to s2			goto s1
s1				
s2				
s3				
s4				



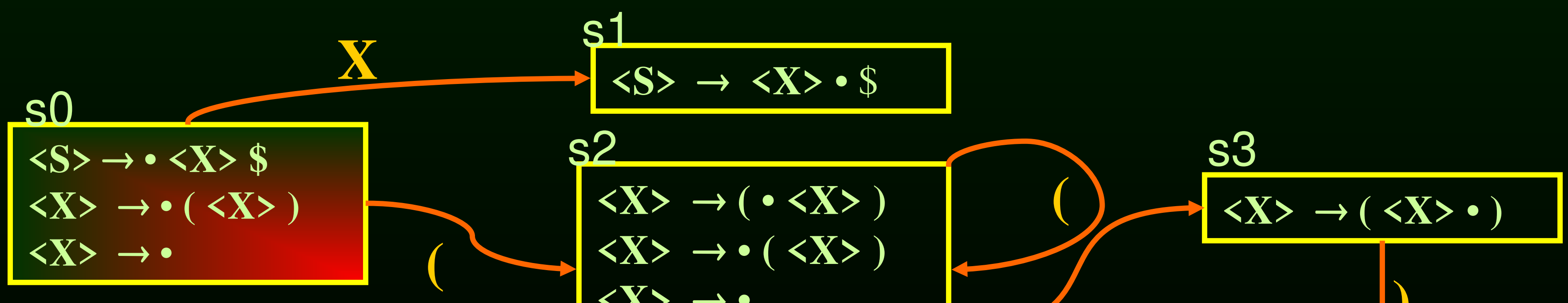
Construcción de la tabla de parseo

	ACTION			Goto
State	(	)	$	X
s0	shift to s2/reduce (3)	reduce (3)	reduce(3)	goto s1
s1				
s2				
s3				
s4				



Construcción de la tabla de parseo

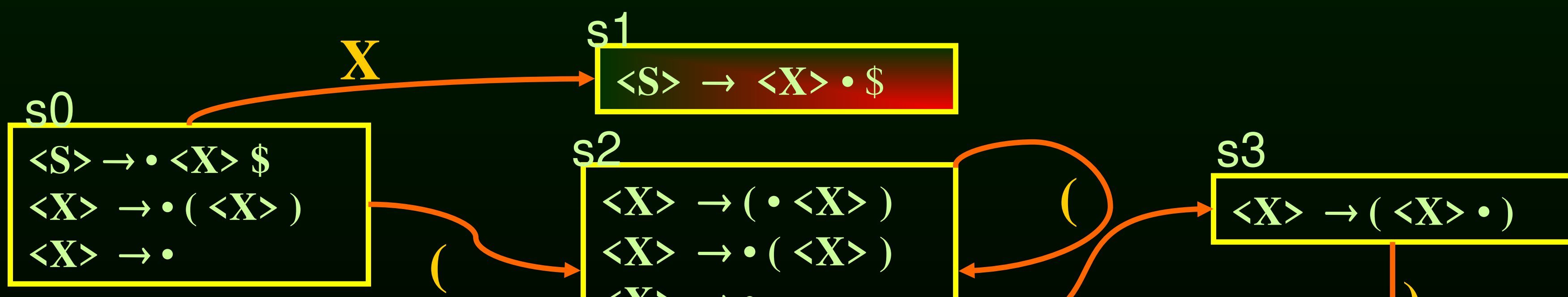
		ACTION		Goto
State	**(**	**)**	**$**	**X**
s0	**shift to s2/reduce (3)**	reduce (3)	reduce(3)	goto s1
s1				
s2				
s3				
s4				



Conflicto shift/reduce

Construcción de la tabla de parseo

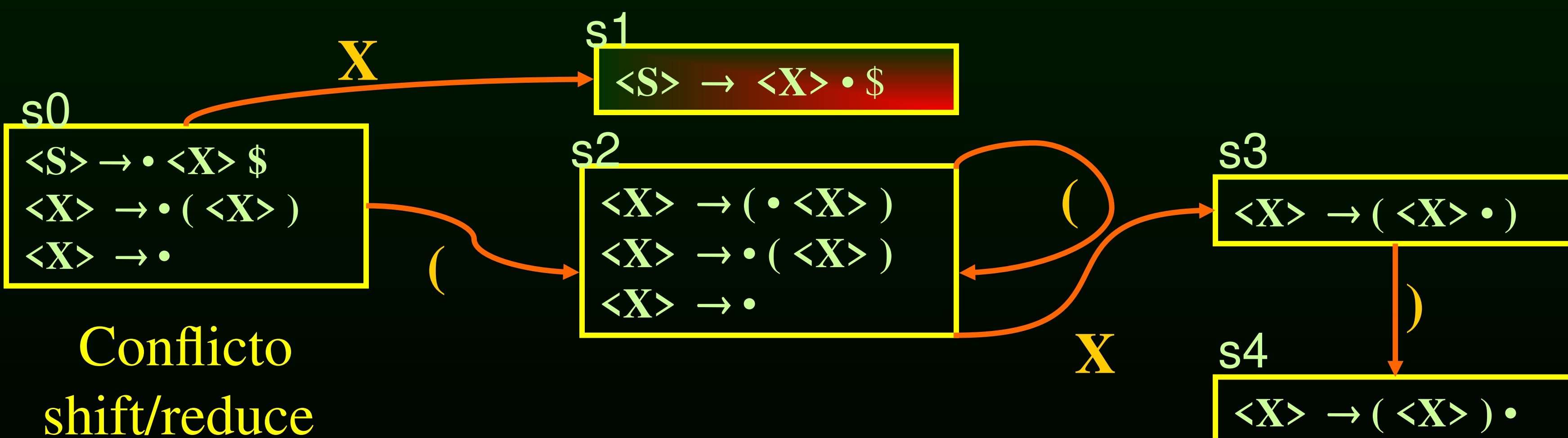
	ACTION			Goto
State	(	)	$	X
s0	shift to s2/reduce (3)	reduce (3)	reduce(3)	goto s1
s1				
s2				
s3				
s4				



Conflicto shift/reduce

Construcción de la tabla de parseo

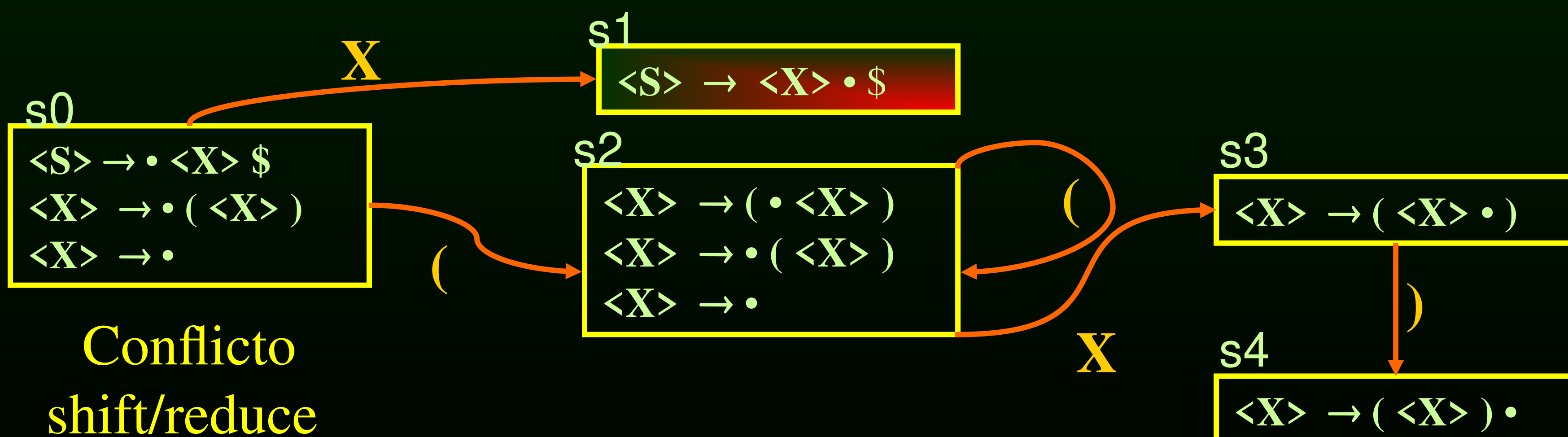
	ACTION			Goto
State	(	)	$	X
s0	shift to s2/reduce (3)	reduce (3)	reduce(3)	goto s1
s1			accept	
s2				
s3				
s4				



Conflicto shift/reduce

Construcción de la tabla de parseo

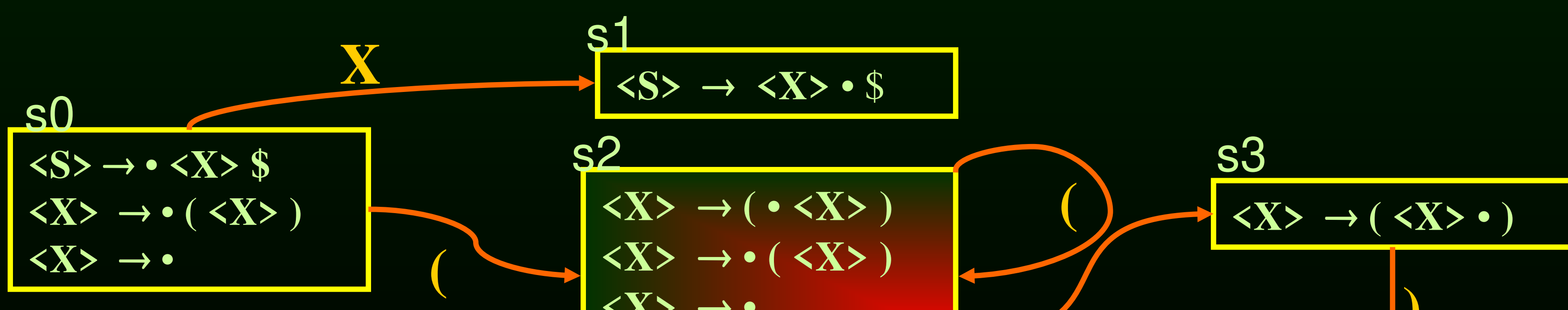
	ACTION			Goto
State	**(**	**)**	**$**	**X**
s0	**shift to s2/reduce (3)**	reduce (3)	reduce(3)	goto s1
s1	error	error	accept	
s2				
s3				
s4				



Conflicto shift/reduce

Construcción de la tabla de parseo

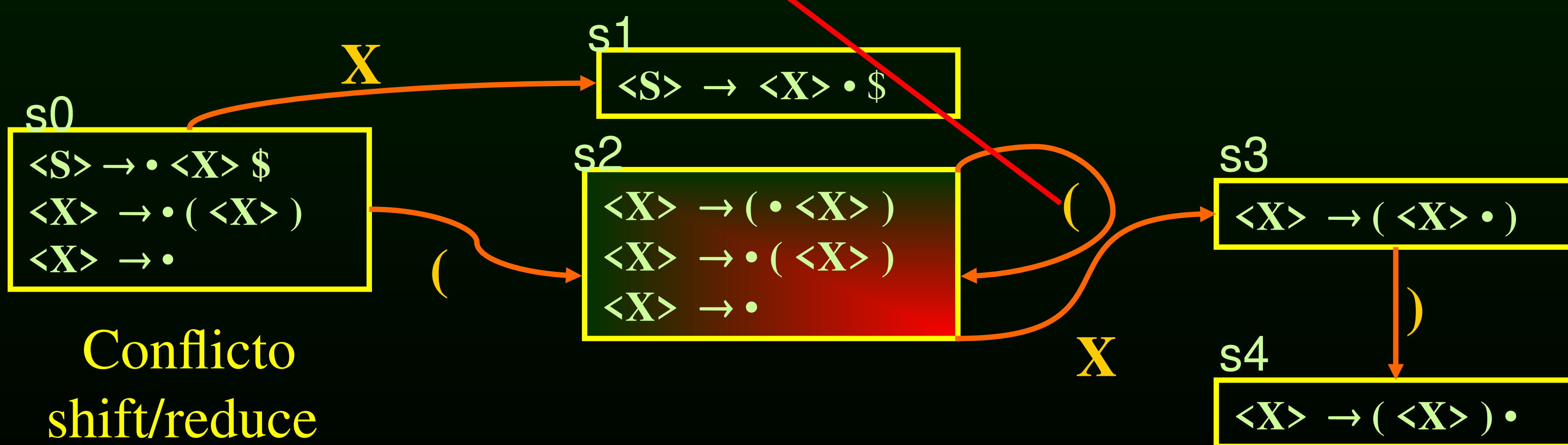
	ACTION			Goto
State	**(**	**)**	**$**	**X**
s0	**shift to s2/reduce (3)**	reduce (3)	reduce(3)	goto s1
s1	error	error	accept	
s2				
s3				
s4				



Conflicto shift/reduce

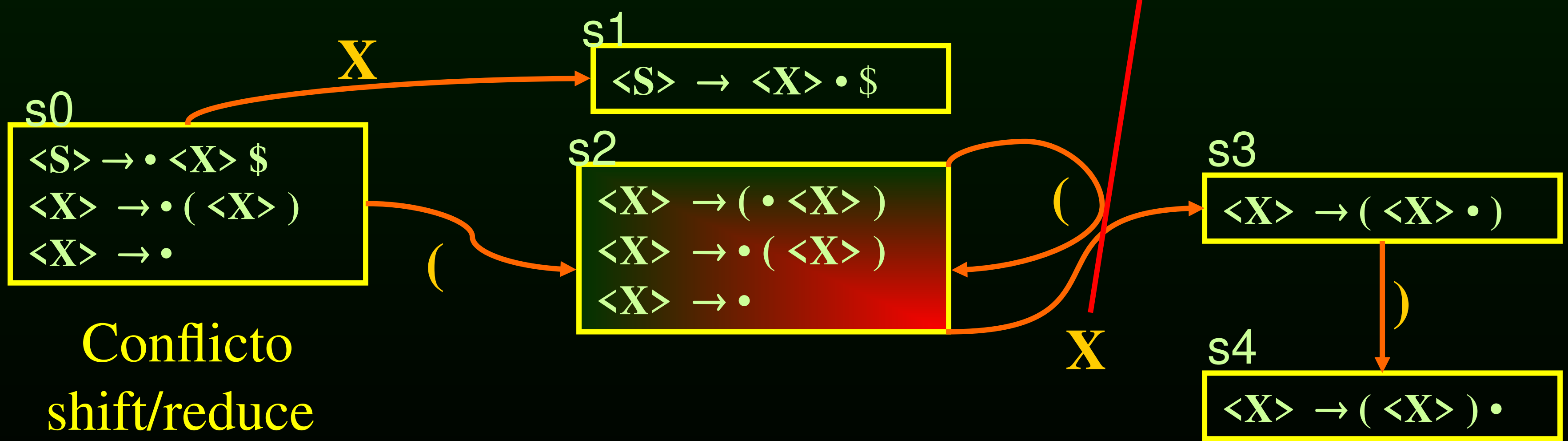
Construcción de la tabla de parseo

	ACTION			Goto
State	(	)	$	X
s0	shift to s2/reduce (3)	reduce (3)	reduce(3)	goto s1
s1	error	error	accept	
s2	shift to s2			
s3				
s4				



Construcción de la tabla de parseo

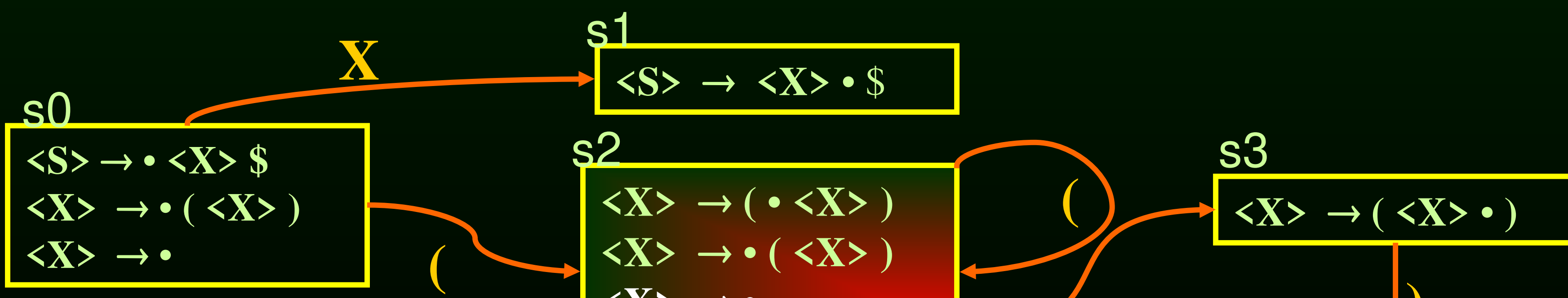
	ACTION			Goto
State	(	)	$	X
s0	shift to s2/reduce (3)	reduce (3)	reduce(3)	goto s1
s1	error	error	accept	
s2	shift to s2			goto s3
s3				
s4				



Conflicto shift/reduce

Construcción de la tabla de parseo

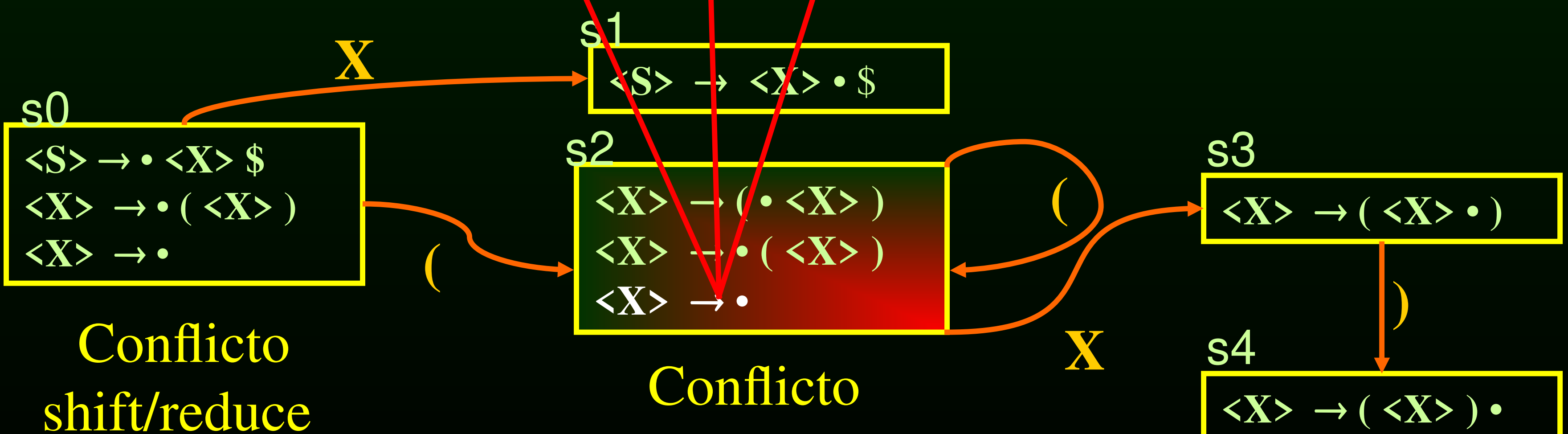
	ACTION			Goto
State	**(**	**)**	**$**	**X**
s0	**shift to s2/reduce (3)**	reduce (3)	reduce(3)	goto s1
s1	error	error	accept	
s2	shift to s2			goto s3
s3				
s4				



Conflicto
shift/reduce

Construcción de la tabla de parseo

	ACTION			Goto
State	(	)	$	X
s0	shift to s2/reduce (3)	reduce (3)	reduce(3)	goto s1
s1	error	error	accept	
s2	shift to s2/reduce (3)	reduce (3)	reduce(3)	goto s3
s3				
s4				

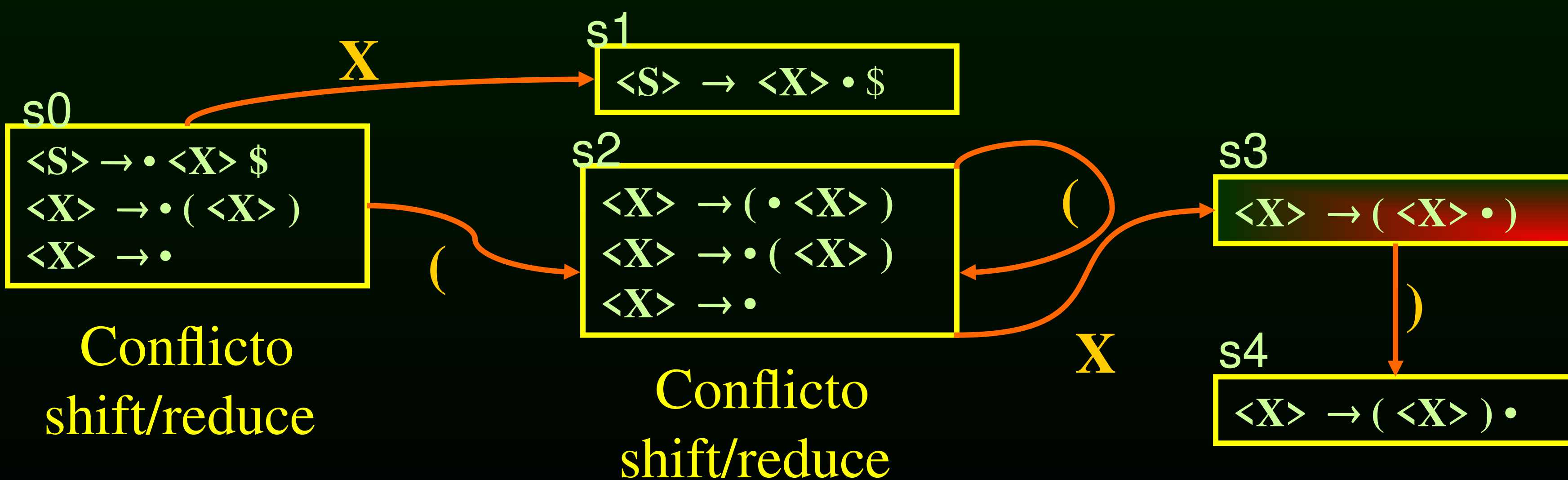


Conflicto shift/reduce

Conflicto shift/reduce

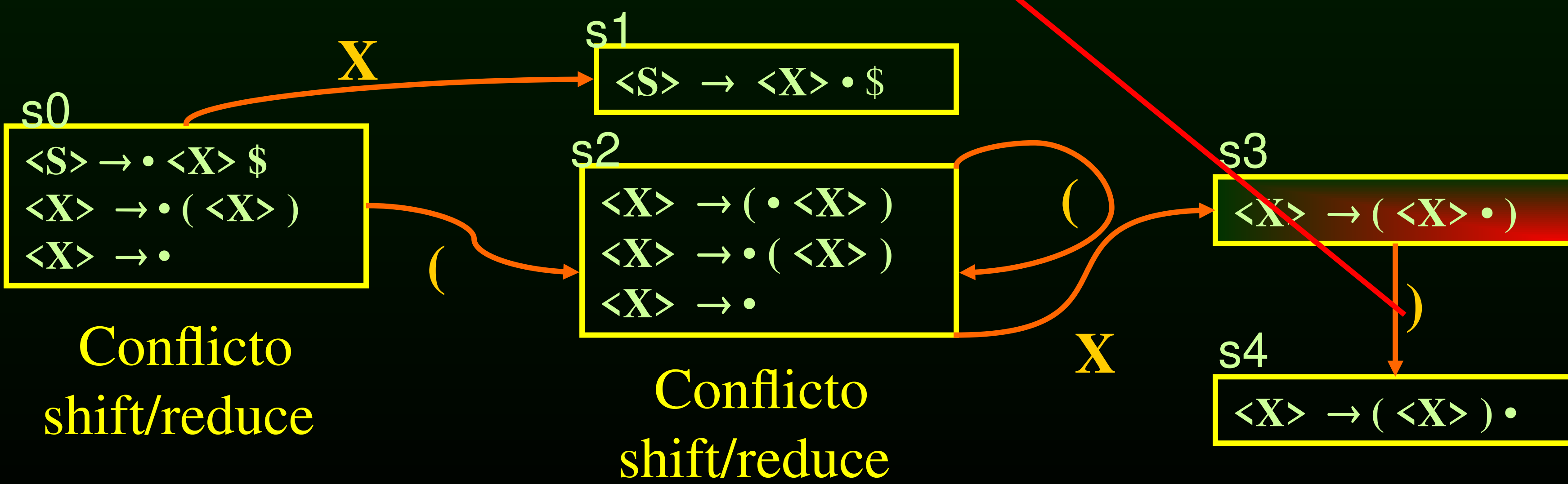
Construcción de la tabla de parseo

		ACTION		Goto
State	**(**	**)**	**$**	**X**
s0	**shift to s2/reduce (3)**	reduce (3)	reduce(3)	goto s1
s1	error	error	accept	
s2	**shift to s2/reduce (3)**	reduce (3)	reduce(3)	goto s3
s3				
s4				



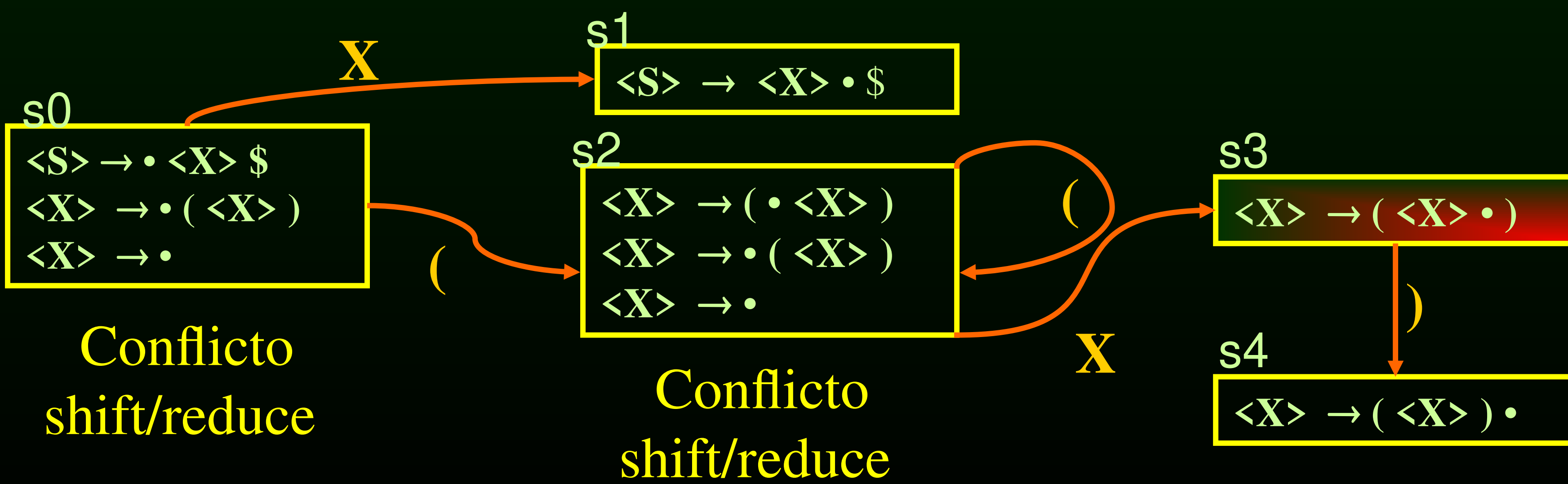
Construcción de la tabla de parseo

State	(	ACTION)	$	Goto X
s0	shift to s2/reduce (3)	reduce (3)	reduce(3)	goto s1
s1	error	error	accept	
s2	shift to s2/reduce (3)	reduce (3)	reduce(3)	goto s3
s3		shift to s4		
s4				



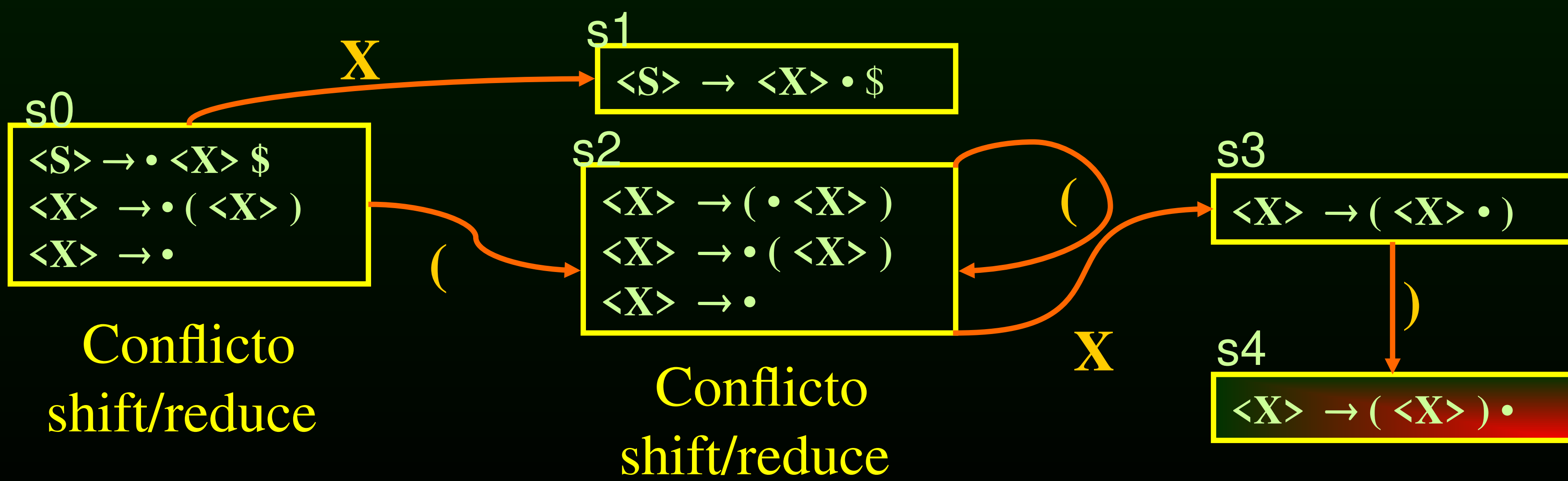
Construcción de la tabla de parseo

		ACTION		Goto
State	(	)	$	X
s0	shift to s2/reduce (3)	reduce (3)	reduce(3)	goto s1
s1	error	error	accept	
s2	shift to s2/reduce (3)	reduce (3)	reduce(3)	goto s3
s3	error	shift to s4	error	
s4				



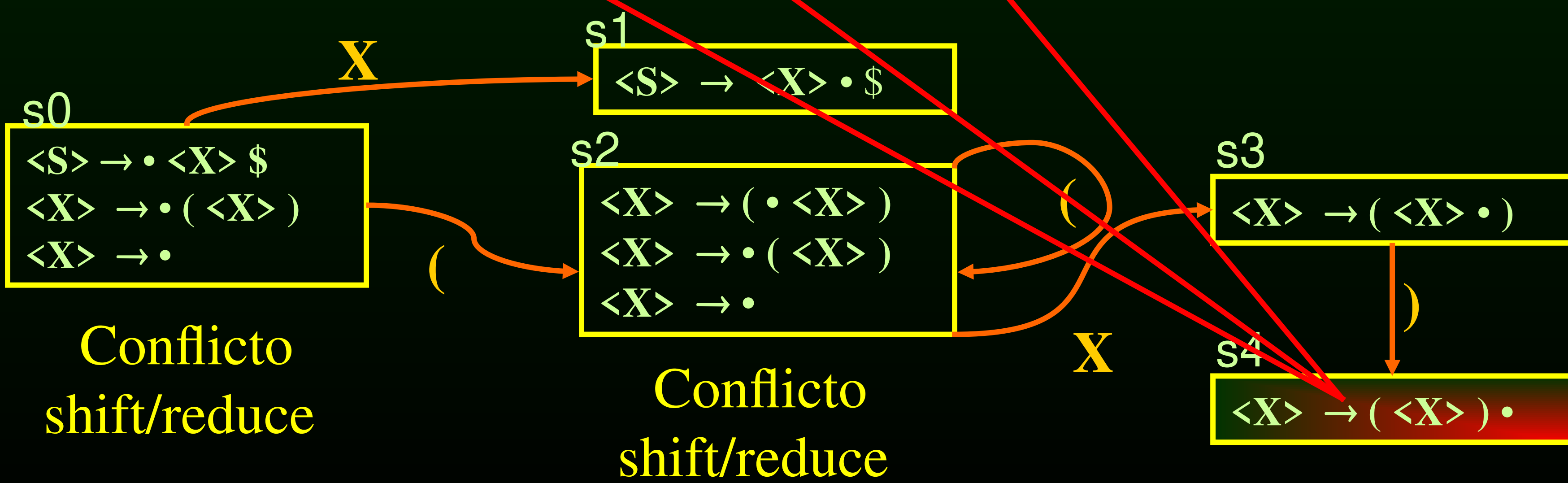
Construcción de la tabla de parseo

	ACTION			Goto
State	(	)	$	X
s0	shift to s2/reduce (3)	reduce (3)	reduce(3)	goto s1
s1	error	error	accept	
s2	shift to s2/reduce (3)	reduce (3)	reduce(3)	goto s3
s3	error	shift to s4	error	
s4				



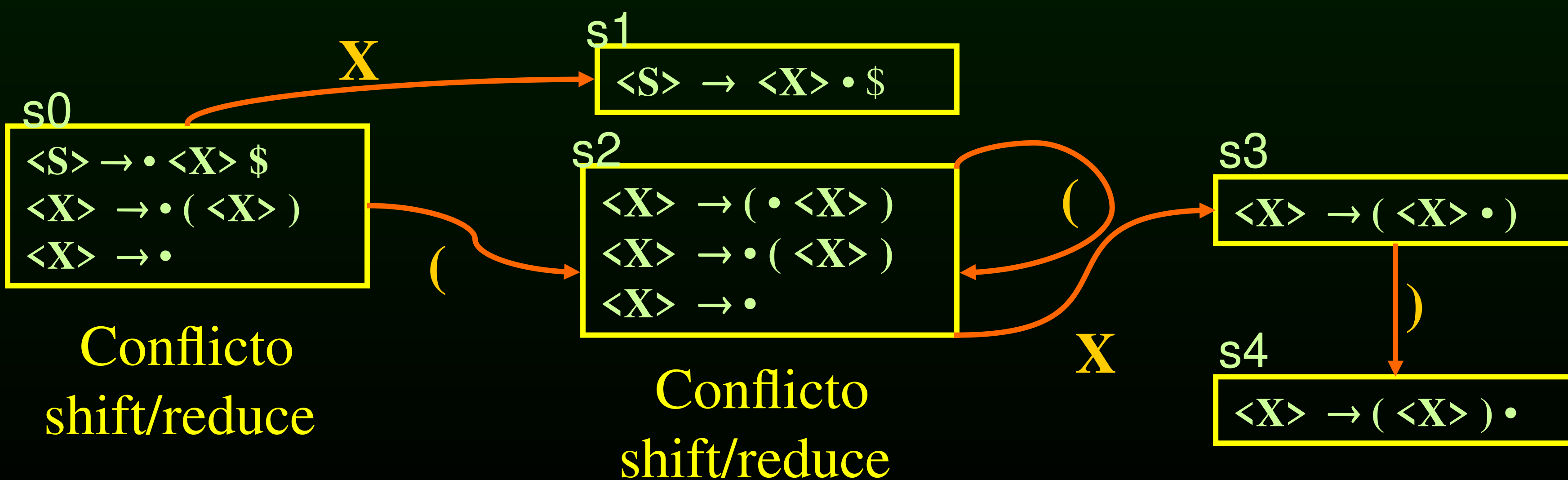
Construcción de la tabla de parseo

State	((ACTION)	) (ACTION)	$ (ACTION)	X (Goto)
s0	shift to s2/reduce (3)	reduce (3)	reduce(3)	goto s1
s1	error	error	accept	
s2	shift to s2/reduce (3)	reduce (3)	reduce(3)	goto s3
s3	error	shift to s4	error	
s4	reduce(2)	reduce(2)	reduce(2)	



Construcción de la tabla de parseo

	ACTION			Goto
State	(	)	$	X
s0	shift to s2/reduce (3)	reduce (3)	reduce(3)	goto s1
s1	error	error	accept	
s2	shift to s2/reduce (3)	reduce (3)	reduce(3)	goto s3
s3	error	shift to s4	error	
s4	reduce(2)	reduce(2)	reduce(2)	



Construcción de la tabla de parseo

	ACTION			Goto
State	(	)	$	X
s0	shift to s2/reduce (3)	reduce (3)	reduce(3)	goto s1
s1	error	error	accept	
s2	shift to s2/reduce (3)	reduce (3)	reduce(3)	goto s3
s3	error	shift to s4	error	
s4	reduce(2)	reduce(2)	reduce(2)	

¿Cómo nos libramos de estos conflictos shift/reduce?

Limitaciones de las gramáticas LR(0)

- Muchos conflictos shift/reduce
- Razón
  - Un ítem $X \rightarrow \alpha \bullet$ en el estado actual identifica una reducción
  - Pero no selecciona cuándo reducir
  - Por lo tanto, tenemos que efectuar la reducción en todos los símbolos de entrada

Resumen

- Resumen de la construcción de un parser LR(0)
- Limitaciones de los lenguajes LR(0)
- Construcción de un parser engine SLR(1)
- Limitaciones de los lenguajes SLR(1)
- Construcción de un parser engine LR(1)
- Construcción de un parser engine LALR(1)

Idea detrás de las gramáticas SLR(1)

- Muchos conflictos shift/reduce en LR(0)
  - un item $X \rightarrow \alpha \bullet$ en el estado actual identifica una reducción
  - Pero no selecciona cuándo reducir
  - Por lo tanto, tenemos que efectuar la reducción en todos los símbolos de entrada
- Hacemos la reducción sólo cuando el símbolo de entrada en efecto sigue a la reducción
  - Tenemos que calcular los terminales que pueden seguir un símbolo no-terminal

follow()

- Para cada no-terminal A, **follow(A)** es el conjunto de terminales que pueden venir después de A

$$\text{follow}(A) = \left\{ b \middle| \quad \exists \text{derivacion} \quad S \Rightarrow \alpha A b \beta \right\}$$

- Si hay una secuencia de derivaciones de la forma S ⇒ α A $ entonces **follow(A)** va a incluir $

Ejemplo de follow()

- Encontrar **follow(<X>)**

  { **???** }

- gramática

  <S> → **<X> $**

  <X> → **(<X>)**

  <X> → ε

Ejemplo de follow()

- Encontrar **follow(<X>)**

  {) }

- gramática

  <S> → **<X>** $

  <X> → (**<X>**)

  <X> → ε

Ejemplo de follow()

- Encontrar **follow(<X>)**

  {) }

- gramática

  <S> → **<X> $**

  <X> → (**<X>**)

  <X> → ε

Ejemplo de follow()

- Encontrar **follow(<X>)**

  {), $ }

- gramática

  <S> $\rightarrow$ **<X> $**

  <X> $\rightarrow$ (**<X>**)

  <X> $\rightarrow$ ε

Ejemplo de follow()

- Encontrar **follow(<X>)**

  {), $ }

- gramática

  <S> → **<X> $**

  <X> → (**<X>**)

  <X> → ε

Pregunta: Encuentren follow()

- Qué es **follow(<S>)**

  { **???** }

- gramática

  <S> → **<X> $**

  <X> → (**<X>**)

  <X> → ε

Pregunta: Encuentren follow()

- Qué es **follow(<S>)**

  { \$ }

- gramática

  <S> → **<X> \$**

  <X> → (**<X>**)

  <X> → ε

Construyendo un parser engine SLR(1)

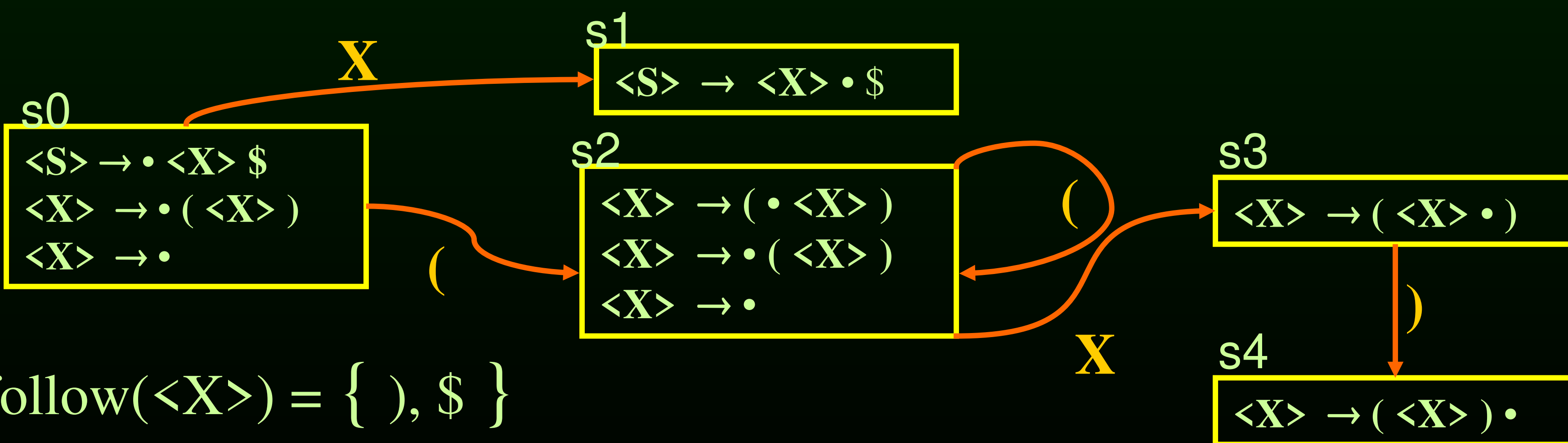
- Agregamos la producción especial **S' → S** $
- Calcular el conjunto **follow** para todos los no-terminales
- Encontrar los ítems de la CFG
- Crear el DFA
  - Usando las funciones **closure** y **goto**
- Construir la tabla de parseo
  - Usando el DFA y la información del conjunto **follow**

Creando las tablas de parseo

- Para cada estado
  - La transición a otro estado usando un símbolo terminal es un shift a ese estado (*shift to sn*)
  - La transición a otro estado usando un símbolo no-terminal es un goto a ese estado (*goto sn*)
  - Si hay un ítem $\mathbf{A \rightarrow \alpha \bullet}$ en el estado, para todos los terminales $\mathbf{c} \in \mathbf{follow(A)}$ hacer una reducción con la producción (*reduce k*)

Construyendo la tabla de parseo

	ACTION			Goto
State	(	)	$	X
s0				
s1				
s2				
s3				
s4				

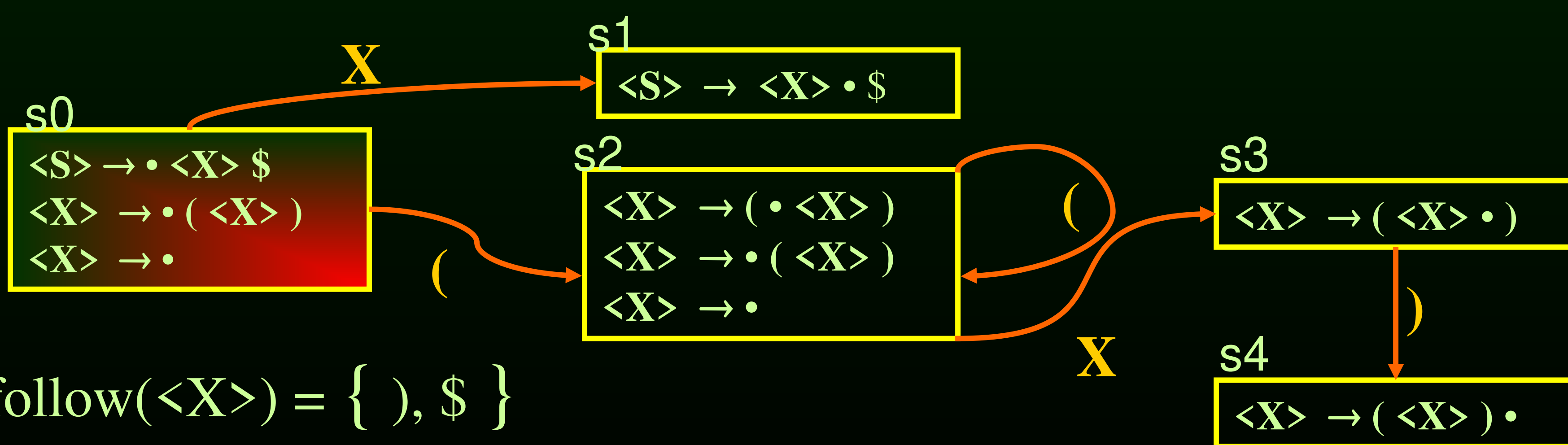


follow(<X>) = {), $ }

follow(<S>) = { $ }

Construyendo la tabla de parseo

	ACTION			Goto
State	(	)	$	X
s0				
s1				
s2				
s3				
s4				

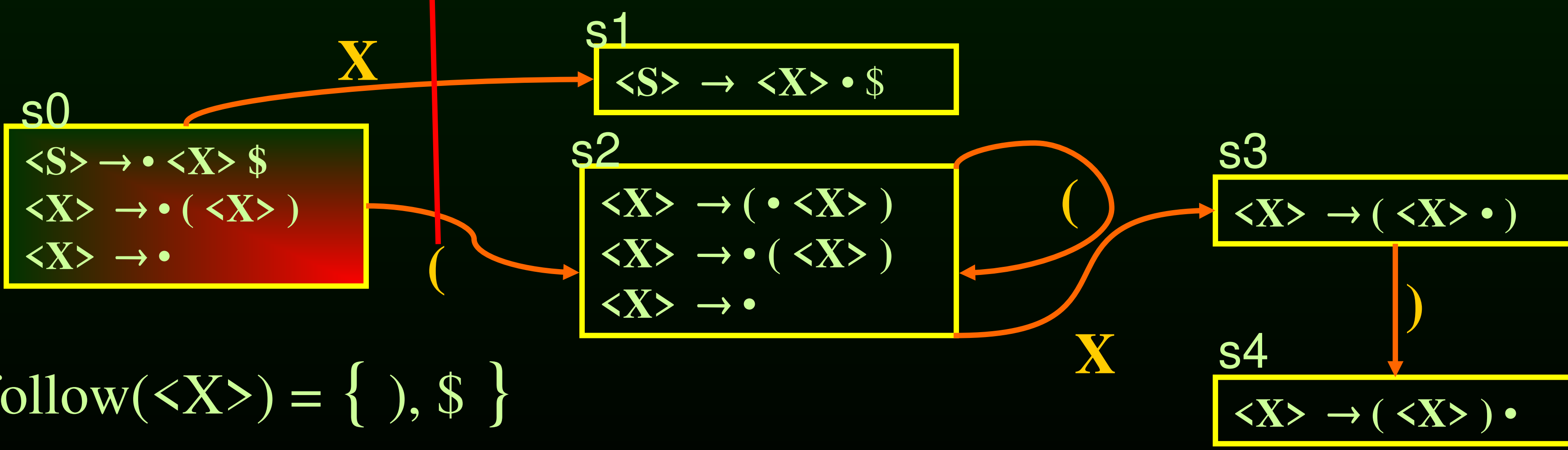


follow(<X>) = {), $ }

follow(<S>) = { $ }

Construyendo la tabla de parseo

State	ACTION (	)	$	Goto X
s0	shift to s2			
s1				
s2				
s3				
s4				

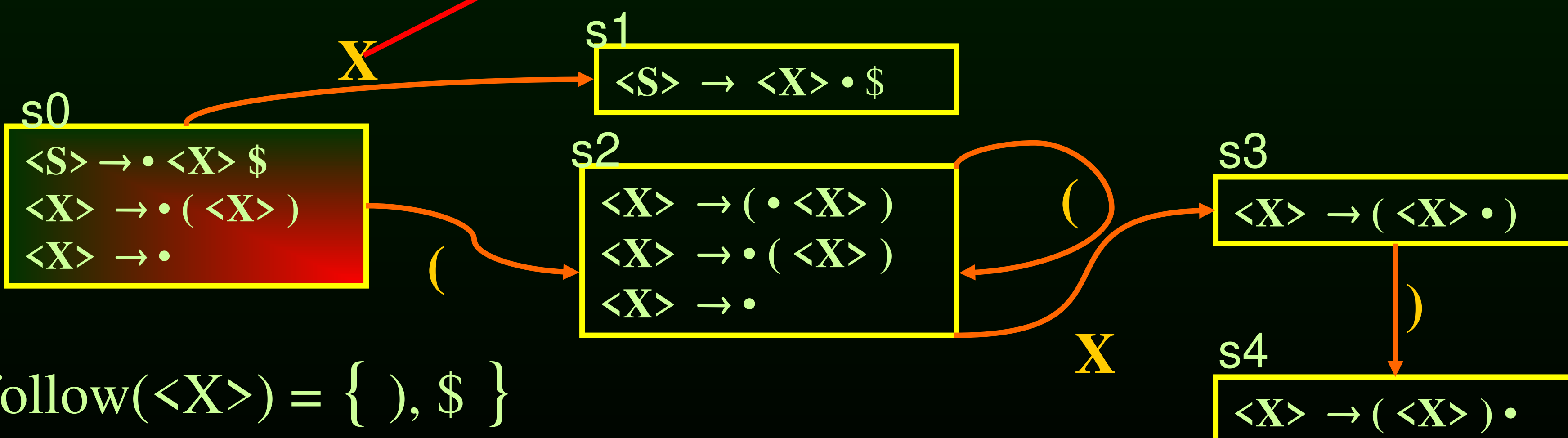


follow(<X>) = {), $ }

follow(<S>) = { $ }

Construyendo la tabla de parseo

	ACTION			Goto
State	(	)	$	X
s0	shift to s2			goto s1
s1				
s2				
s3				
s4				

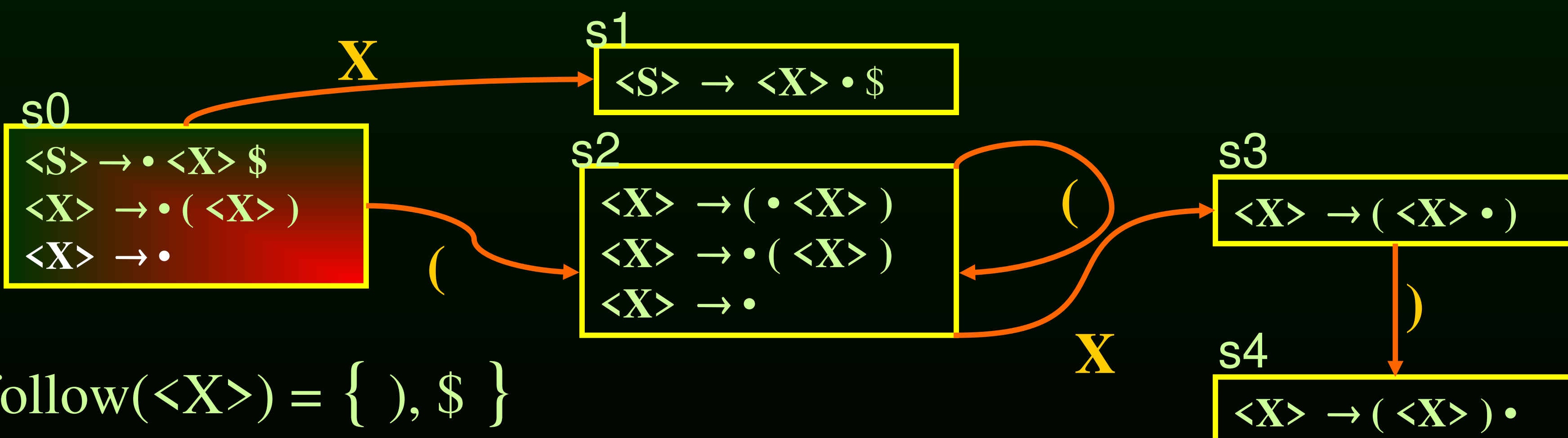


$\text{follow}(\langle X \rangle) = \{\ ),\ \$\ \}$

$\text{follow}(\langle S \rangle) = \{\ \$\ \}$

Construyendo la tabla de parseo

	ACTION			Goto
State	(	)	$	X
s0	shift to s2			goto s1
s1				
s2				
s3				
s4				

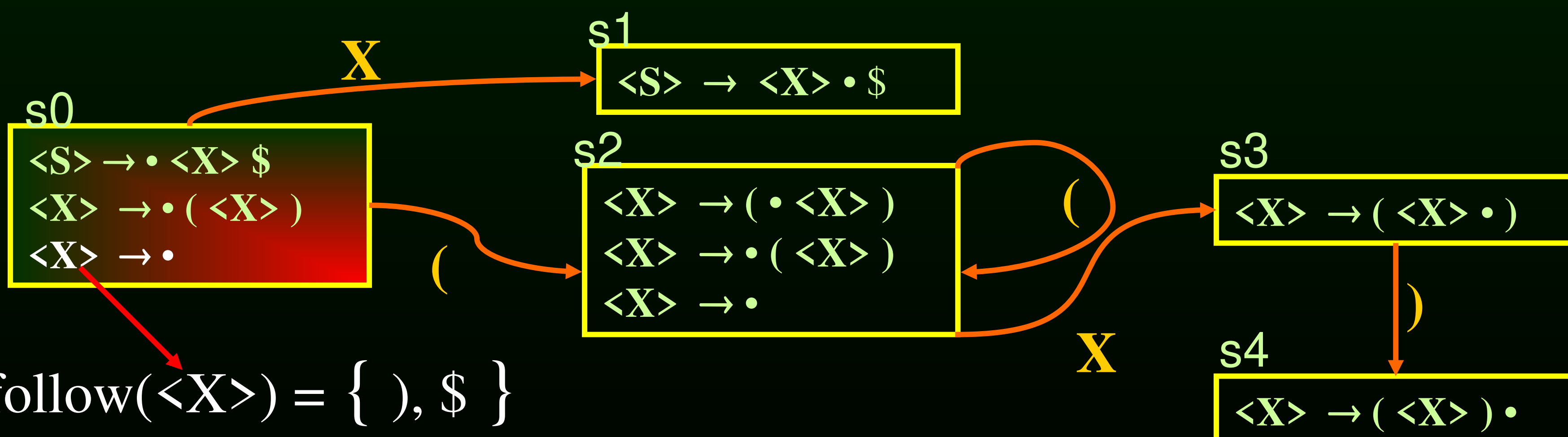


follow(<X>) = {), $ }

follow(<S>) = { $ }

Construyendo la tabla de parseo

	ACTION			Goto
State	(	)	$	X
s0	shift to s2/reduce (3)	reduce (3)	reduce(3)	goto s1
s1				
s2				
s3				
s4				

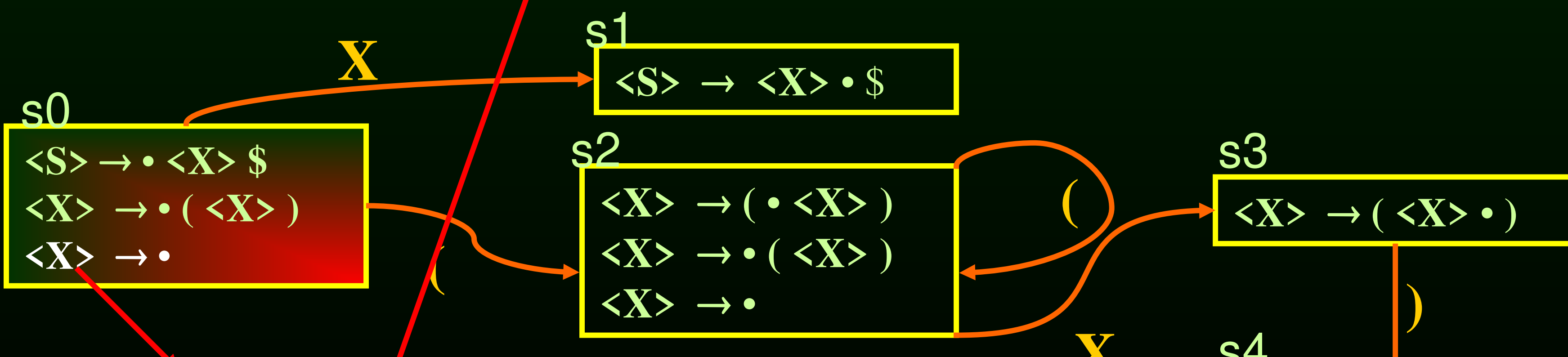


follow(<X>) = {), $ }

follow(<S>) = { $ }

Construyendo la tabla de parseo

		ACTION		Goto
State	**(**	**)**	**$**	**X**
s0	shift to s2	reduce (3)		goto s1
s1				
s2				
s3				
s4				

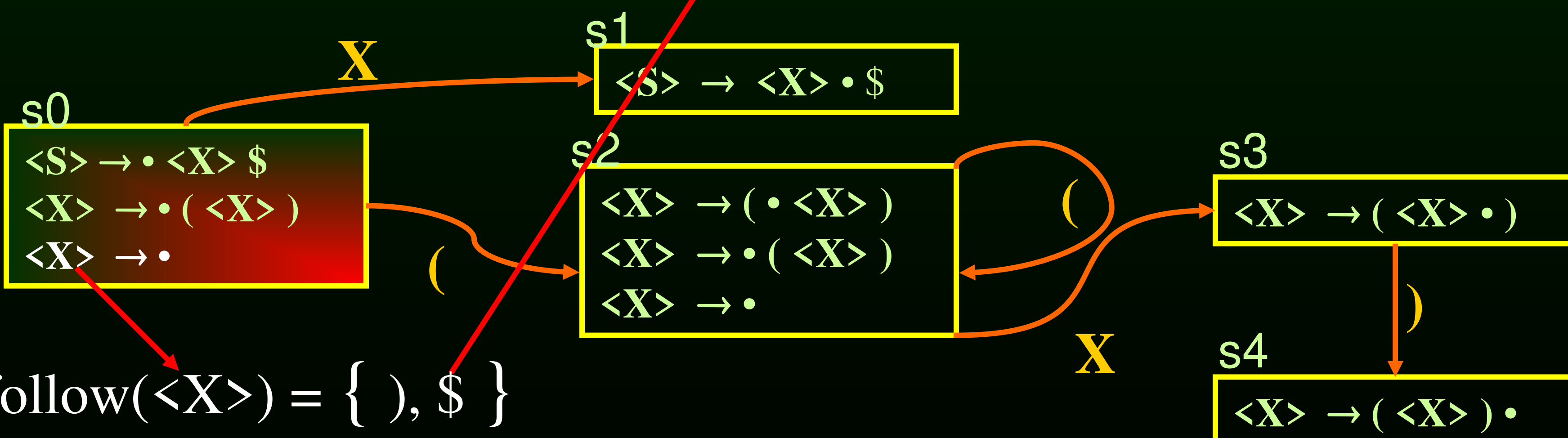


follow(<X>) = {), $ }

follow(<S>) = { $ }

Construyendo la tabla de parseo

	ACTION			Goto
State	(	)	$	X
s0	shift to s2	reduce (3)	reduce (3)	goto s1
s1				
s2				
s3				
s4				

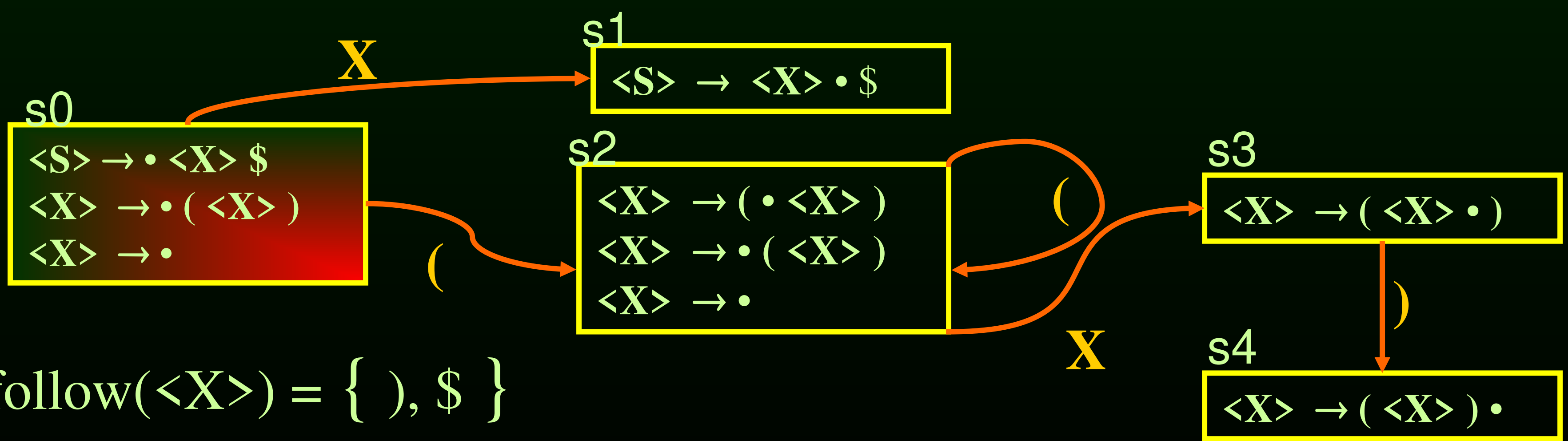


follow(<X>) = {), $ }

follow(<S>) = { $ }

Construyendo la tabla de parseo

	ACTION			Goto
State	**(**	**)**	**$**	**X**
s0	shift to s2	reduce (3)	reduce (3)	goto s1
s1				
s2				
s3				
s4				

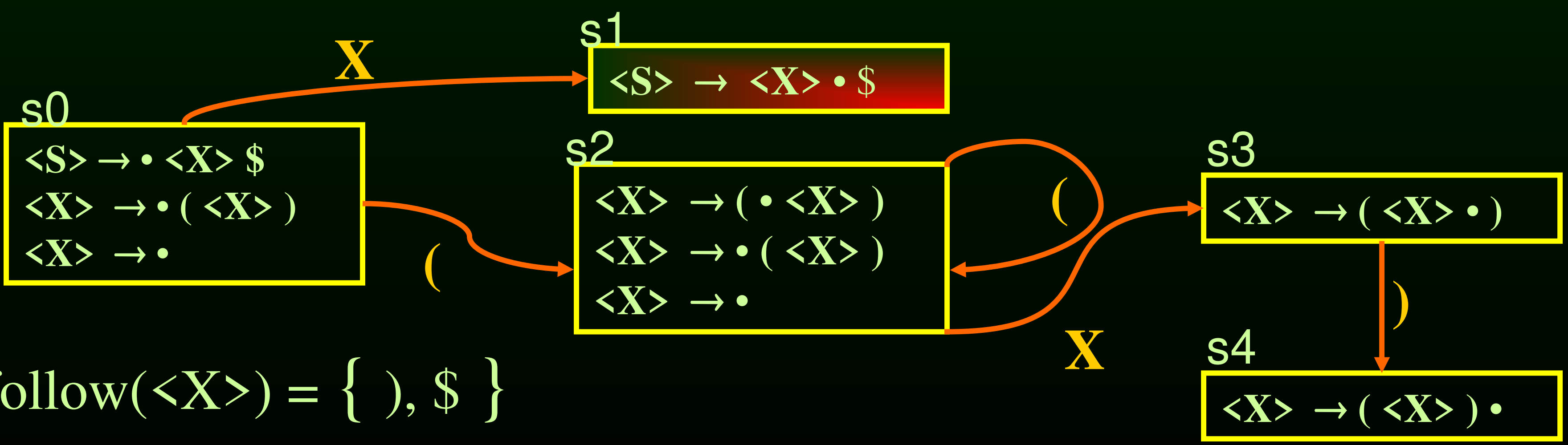


follow(<X>) = {), $ }

follow(<S>) = { $ }

Construyendo la tabla de parseo

	ACTION			Goto
State	(	)	$	X
s0	shift to s2	reduce (3)	reduce (3)	goto s1
s1				
s2				
s3				
s4				

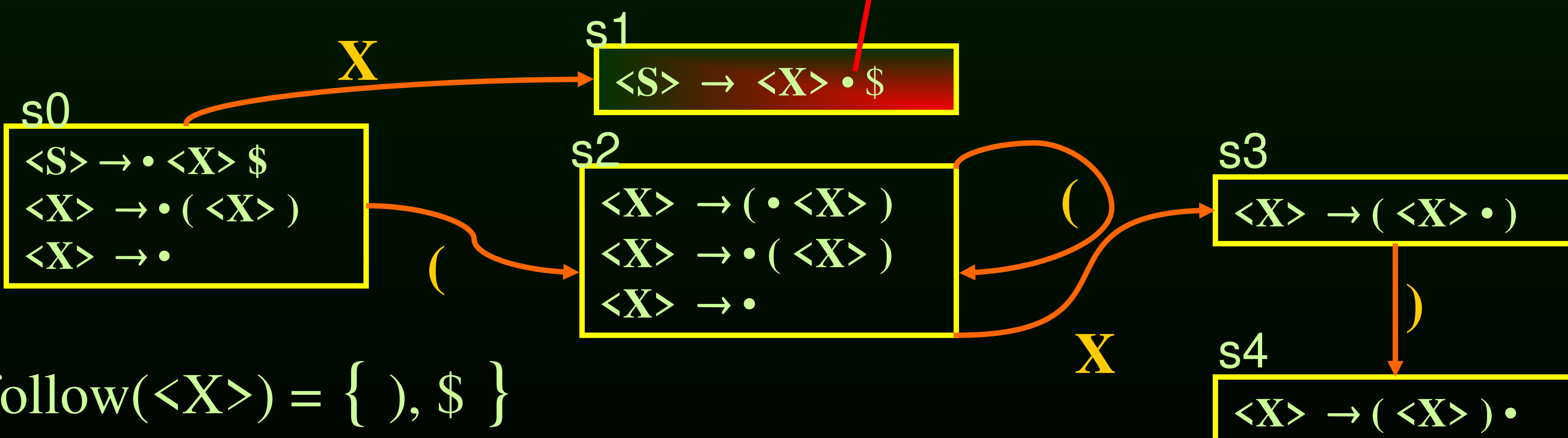


follow(<X>) = {), $ }

follow(<S>) = { $ }

Construyendo la tabla de parseo

	ACTION			Goto
State	**(**	**)**	**$**	**X**
s0	shift to s2	reduce (3)	reduce (3)	goto s1
s1			accept	
s2				
s3				
s4				

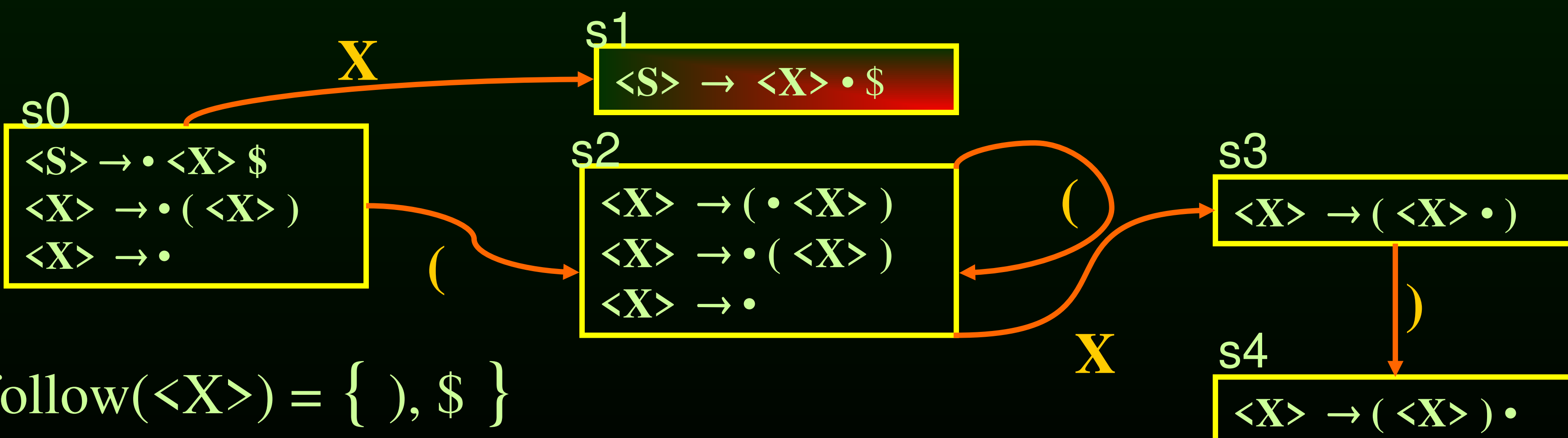


follow(<X>) = {), $ }

follow(<S>) = { $ }

Construyendo la tabla de parseo

	ACTION			Goto
State	**(**	**)**	**$**	**X**
s0	shift to s2	reduce (3)	reduce (3)	goto s1
s1	error	error	accept	
s2				
s3				
s4				

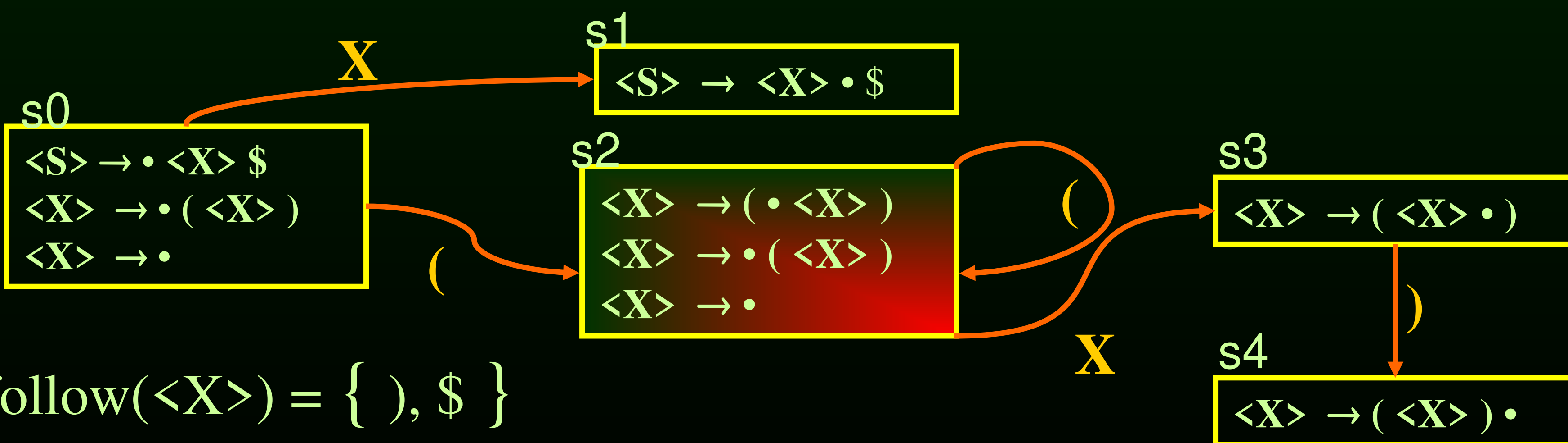


$\text{follow}(\langle X\rangle) = \{\ ),\ \$\ \}$

$\text{follow}(\langle S\rangle) = \{\ \$\ \}$

Construyendo la tabla de parseo

	ACTION			Goto
State	**(**	**)**	**$**	**X**
s0	shift to s2	reduce (3)	reduce (3)	goto s1
s1	error	error	accept	
s2				
s3				
s4				

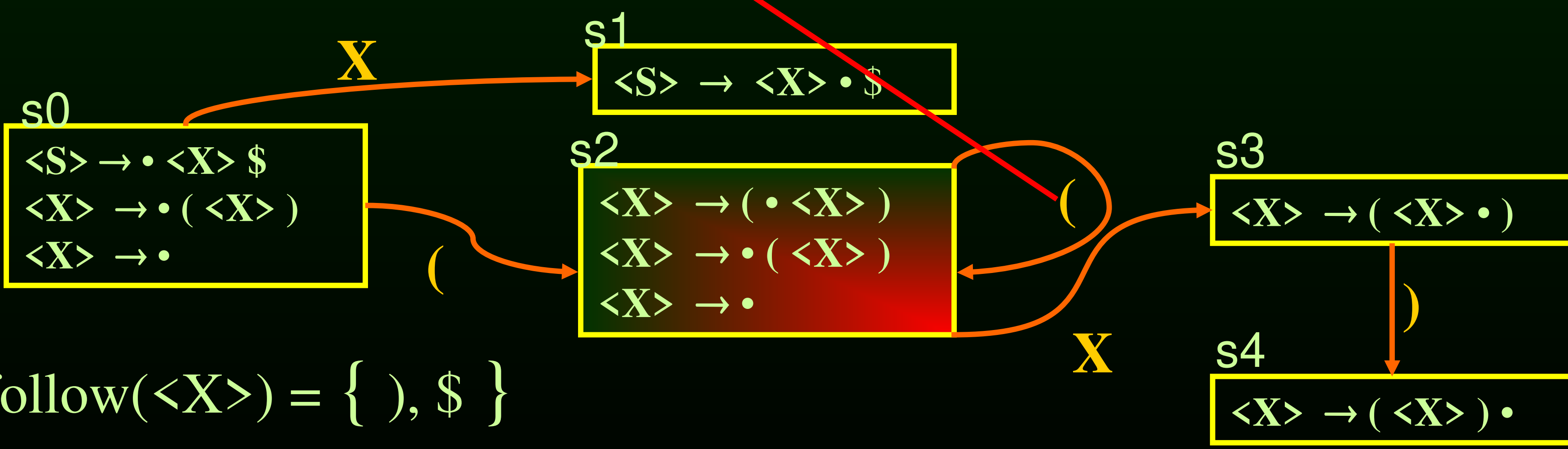


follow(<X>) = {), $ }

follow(<S>) = { $ }

Construyendo la tabla de parseo

	ACTION			Goto
State	(	)	$	X
s0	shift to s2	reduce (3)	reduce (3)	goto s1
s1	error	error	accept	
s2	shift to s2			
s3				
s4				

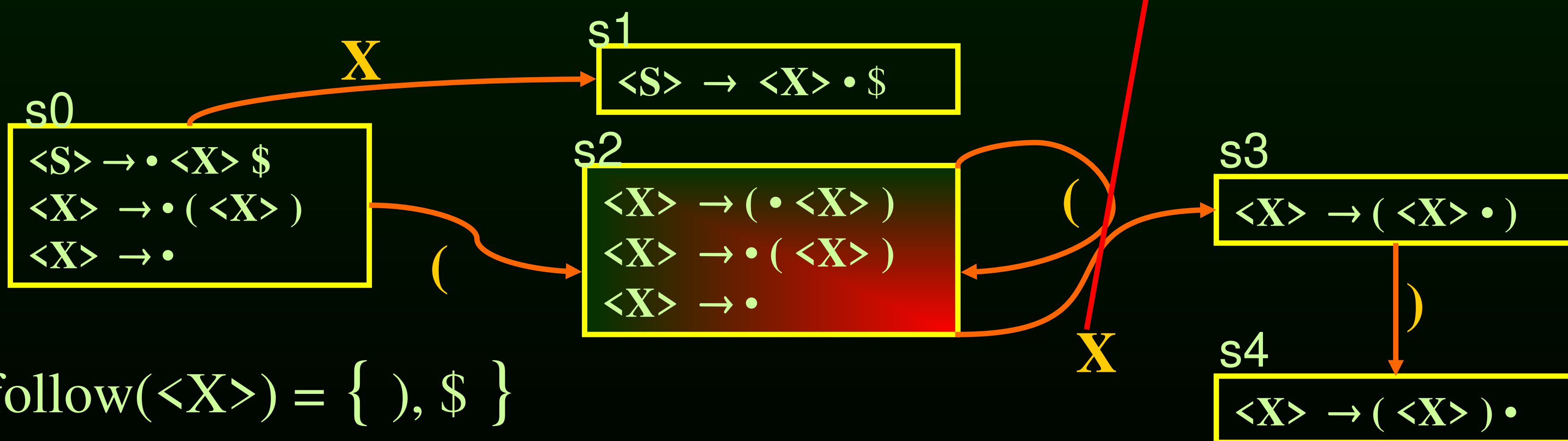


follow(<X>) = {), $ }

follow(<S>) = { $ }

Construyendo la tabla de parseo

State	ACTION (	ACTION)	ACTION $	Goto X
s0	shift to s2	reduce (3)	reduce (3)	goto s1
s1	error	error	accept	
s2	shift to s2			goto s3
s3				
s4				

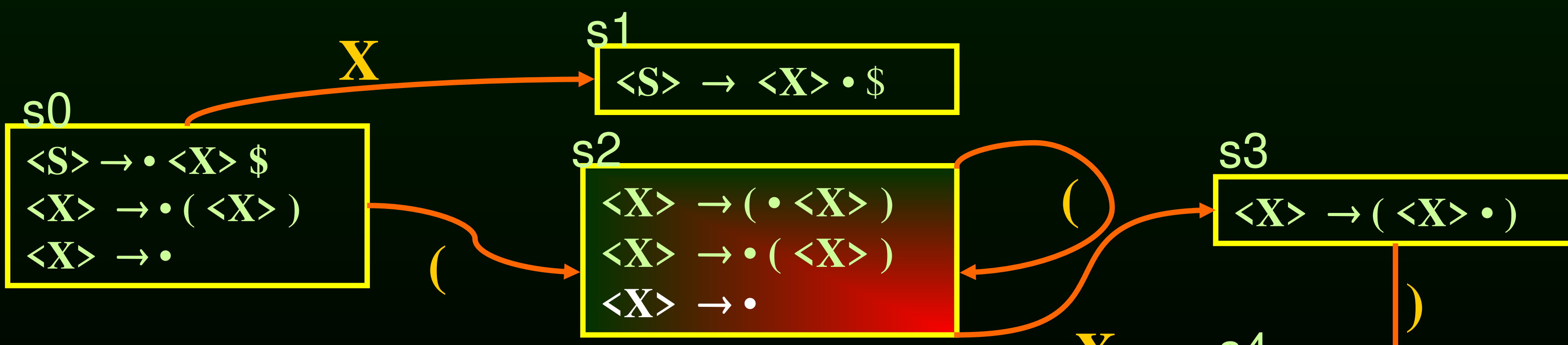


follow(<X>) = {), $ }

follow(<S>) = { $ }

Construyendo la tabla de parseo

		ACTION		Goto
State	(	)	$	X
s0	shift to s2	reduce (3)	reduce (3)	goto s1
s1	error	error	accept	
s2	shift to s2			goto s3
s3				
s4				

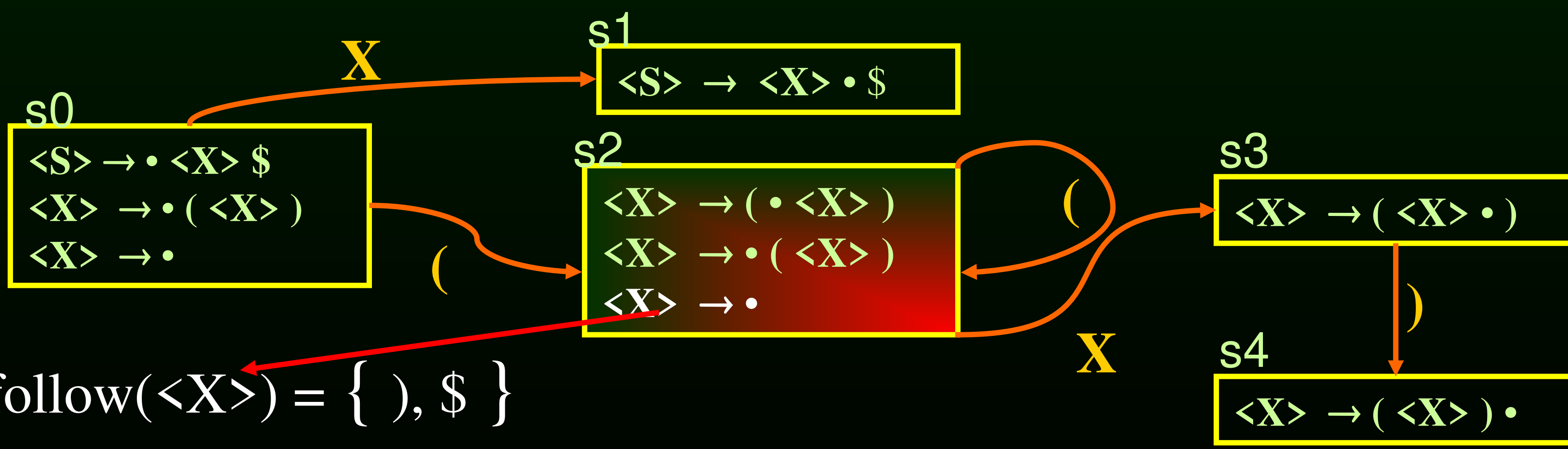


follow(<X>) = {), $ }

follow(<S>) = { $ }

Construyendo la tabla de parseo

		ACTION		Goto
State	(	)	$	X
s0	shift to s2	reduce (3)	reduce (3)	goto s1
s1	error	error	accept	
s2	shift to s2			goto s3
s3				
s4				

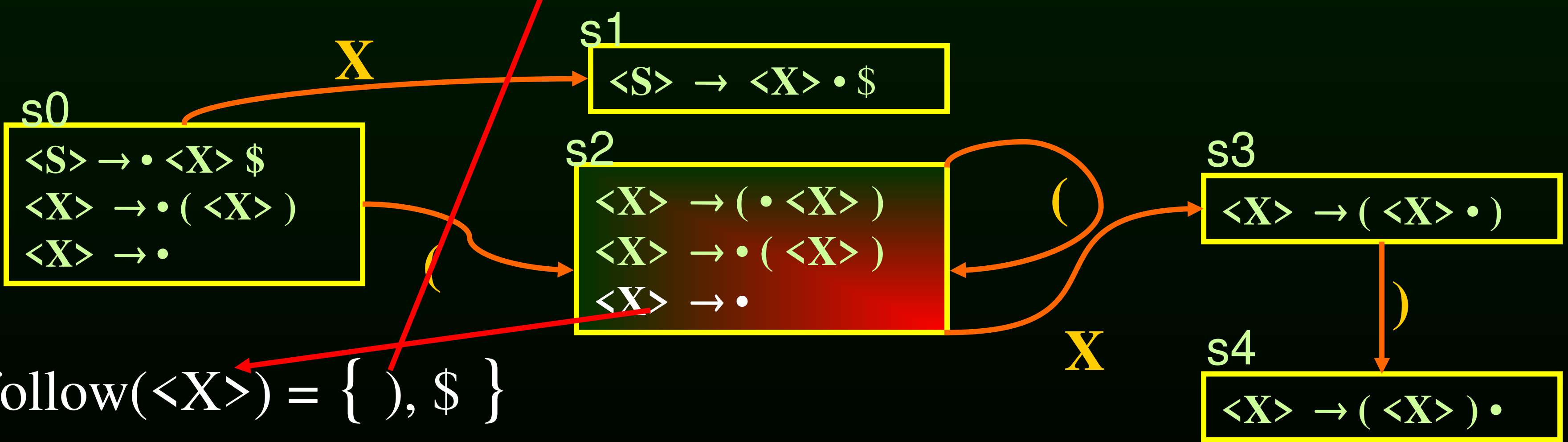


follow(<X>) = {), $ }

follow(<S>) = { $ }

Construyendo la tabla de parseo

	ACTION			Goto
State	(	)	$	X
s0	shift to s2	reduce (3)	reduce (3)	goto s1
s1	error	error	accept	
s2	shift to s2	reduce (3)		goto s3
s3				
s4				

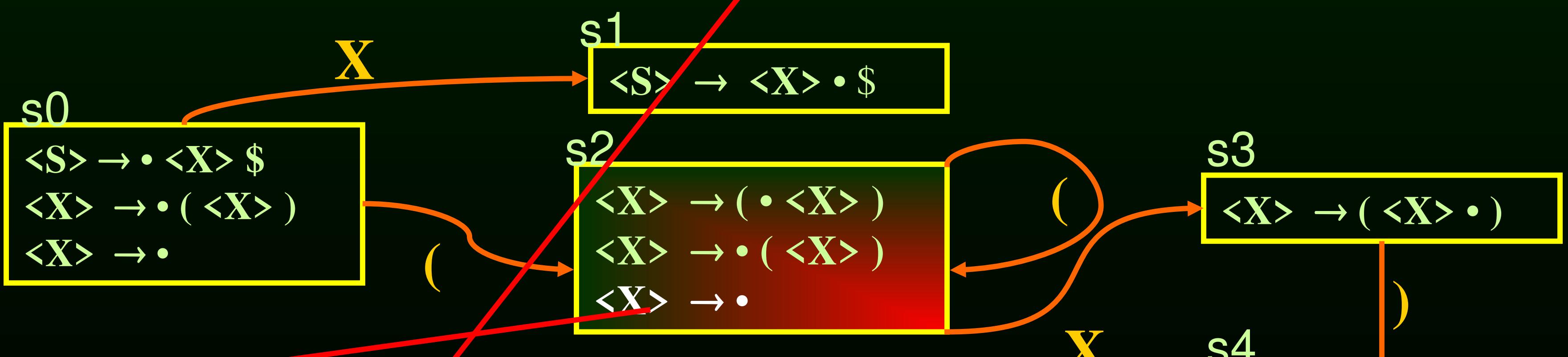


follow(<X>) = {), $ }

follow(<S>) = { $ }

Construyendo la tabla de parseo

State	ACTION			Goto
	(	)	$	X
s0	shift to s2	reduce (3)	reduce (3)	goto s1
s1	error	error	accept	
s2	shift to s2	reduce (3)	reduce(3)	goto s3
s3				
s4				

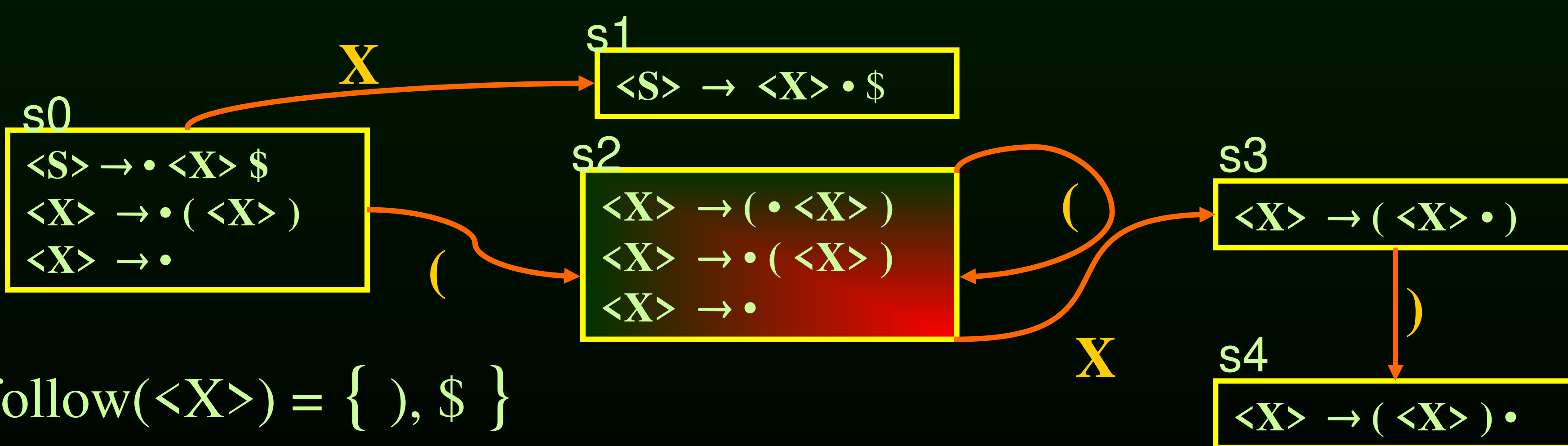


follow(<X>) = {), $ }

follow(<S>) = { $ }

Construyendo la tabla de parseo

	ACTION			Goto
State	(	)	$	X
s0	shift to s2	reduce (3)	reduce (3)	goto s1
s1	error	error	accept	
s2	shift to s2	reduce (3)	reduce(3)	goto s3
s3				
s4				

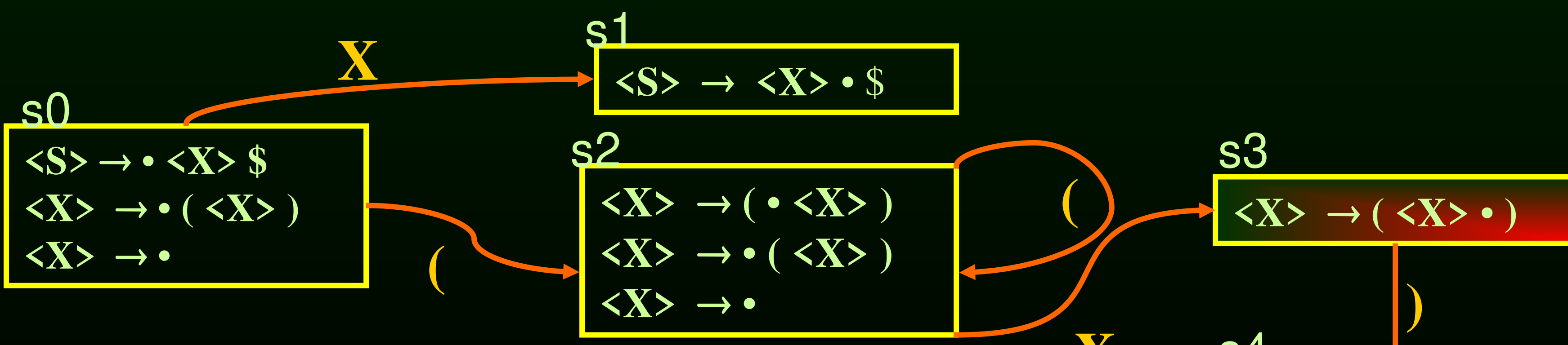


$$\text{follow}(\langle X\rangle) = \{\ ),\ \$\ \}$$

$$\text{follow}(\langle S\rangle) = \{\ \$\ \}$$

Construyendo la tabla de parseo

	ACTION			Goto
State	(	)	$	X
s0	shift to s2	reduce (3)	reduce (3)	goto s1
s1	error	error	accept	
s2	shift to s2	reduce (3)	reduce(3)	goto s3
s3				
s4				

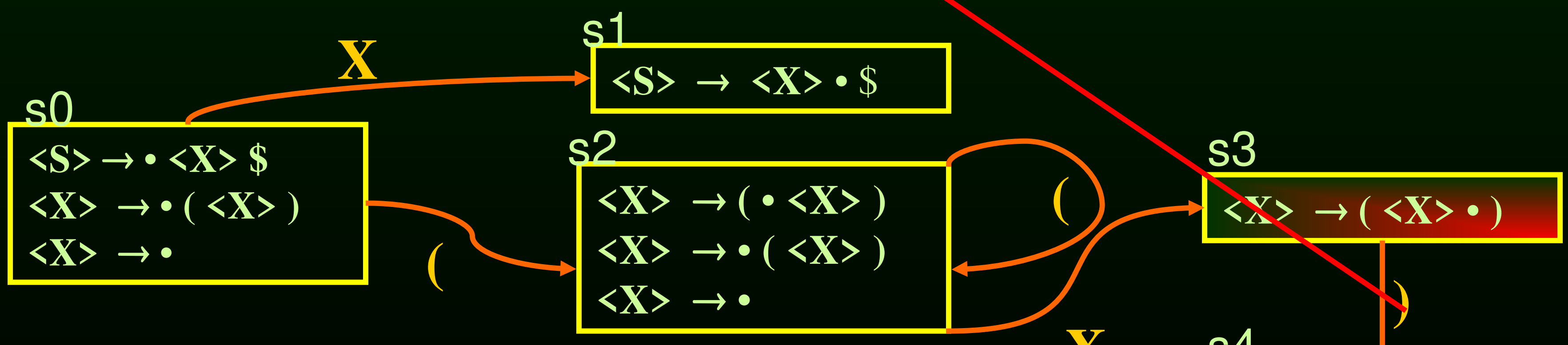


follow(<X>) = {), $ }

follow(<S>) = { $ }

Construyendo la tabla de parseo

	ACTION			Goto
State	(	)	$	X
s0	shift to s2	reduce (3)	reduce (3)	goto s1
s1	error	error	accept	
s2	shift to s2	reduce (3)	reduce(3)	goto s3
s3		shift to s4		
s4				

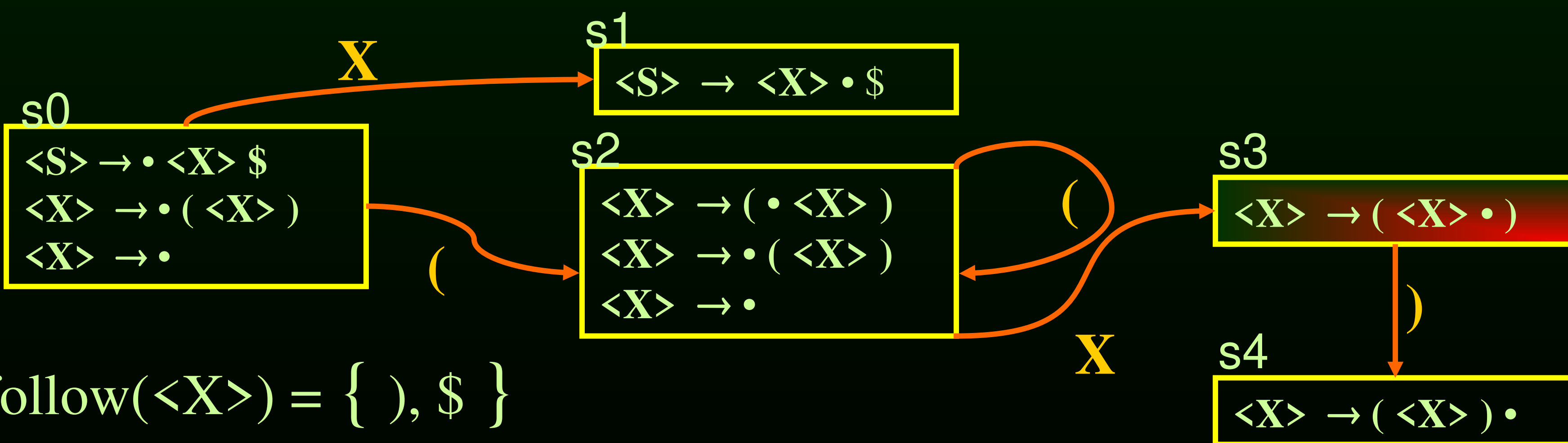


follow(<X>) = {), $ }

follow(<S>) = { $ }

Construyendo la tabla de parseo

State	ACTION (	ACTION)	ACTION $	Goto X
s0	shift to s2	reduce (3)	reduce (3)	goto s1
s1	error	error	accept	
s2	shift to s2	reduce (3)	reduce(3)	goto s3
s3	error	shift to s4	error	
s4				

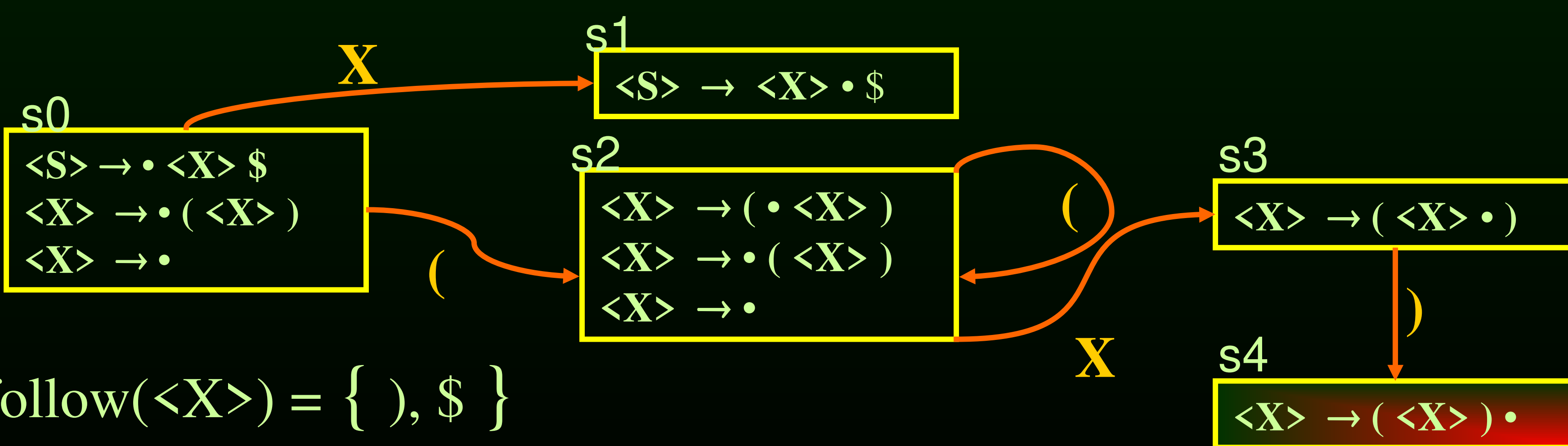


follow(<X>) = {), $ }

follow(<S>) = { $ }

Construyendo la tabla de parseo

	ACTION			Goto
State	(	)	$	X
s0	shift to s2	reduce (3)	reduce (3)	goto s1
s1	error	error	accept	
s2	shift to s2	reduce (3)	reduce(3)	goto s3
s3	error	shift to s4	error	
s4				

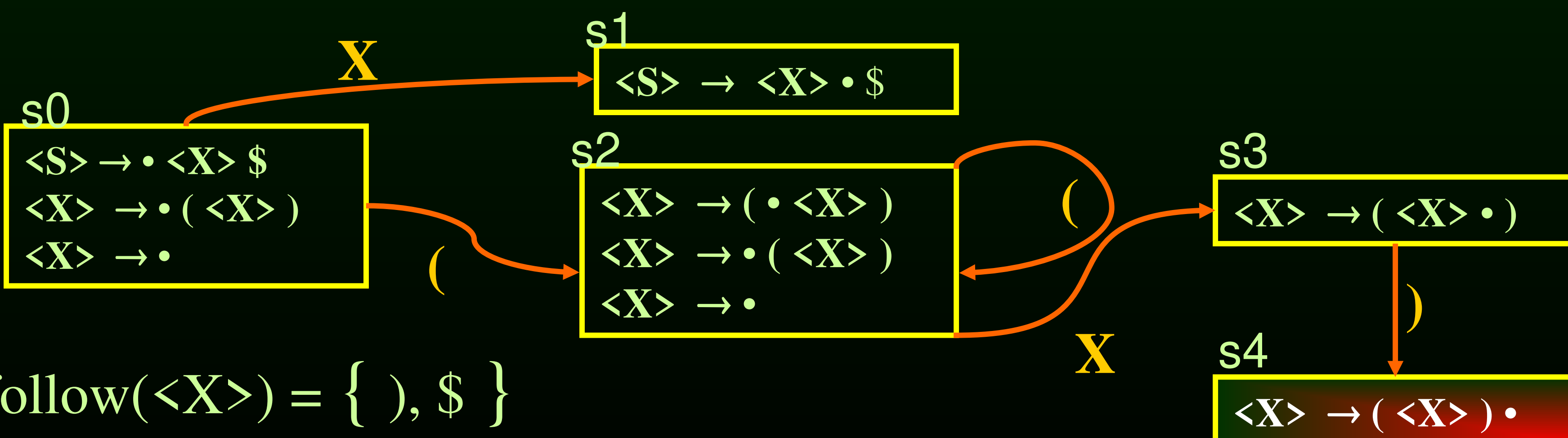


$$\text{follow}(\langle X\rangle) = \{\ ),\ \$\ \}$$

$$\text{follow}(\langle S\rangle) = \{\ \$\ \}$$

Construyendo la tabla de parseo

	ACTION			Goto
State	**(**	**)**	**$**	**X**
s0	shift to s2	reduce (3)	reduce (3)	goto s1
s1	error	error	accept	
s2	shift to s2	reduce (3)	reduce(3)	goto s3
s3	error	shift to s4	error	
s4				

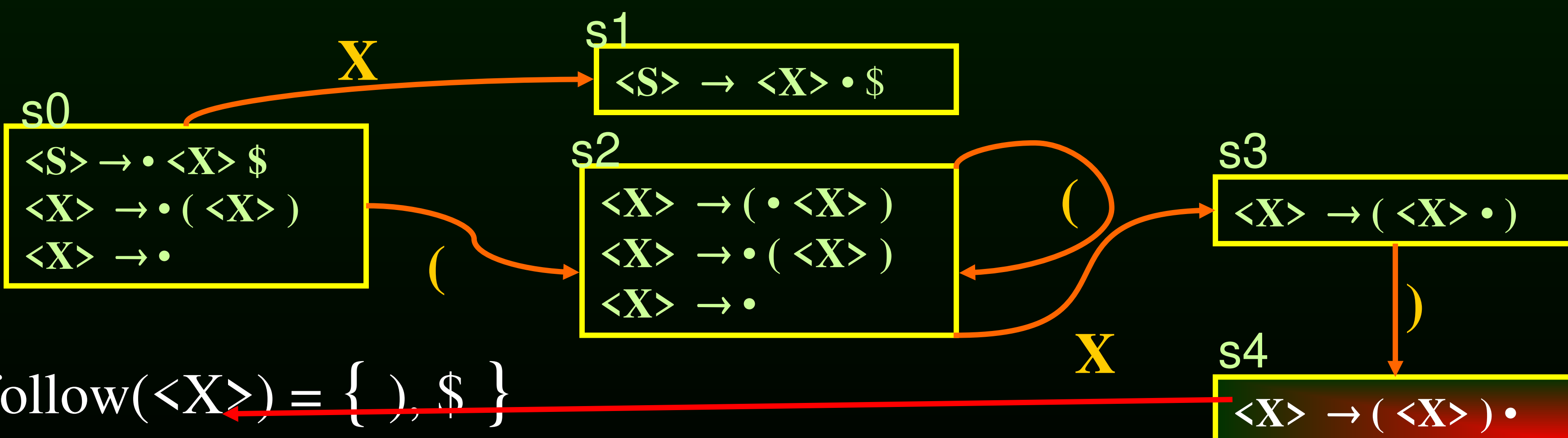


follow(<X>) = {), $ }

follow(<S>) = { $ }

Construyendo la tabla de parseo

	ACTION			Goto
State	(	)	$	X
s0	shift to s2	reduce (3)	reduce (3)	goto s1
s1	error	error	accept	
s2	shift to s2	reduce (3)	reduce(3)	goto s3
s3	error	shift to s4	error	
s4				

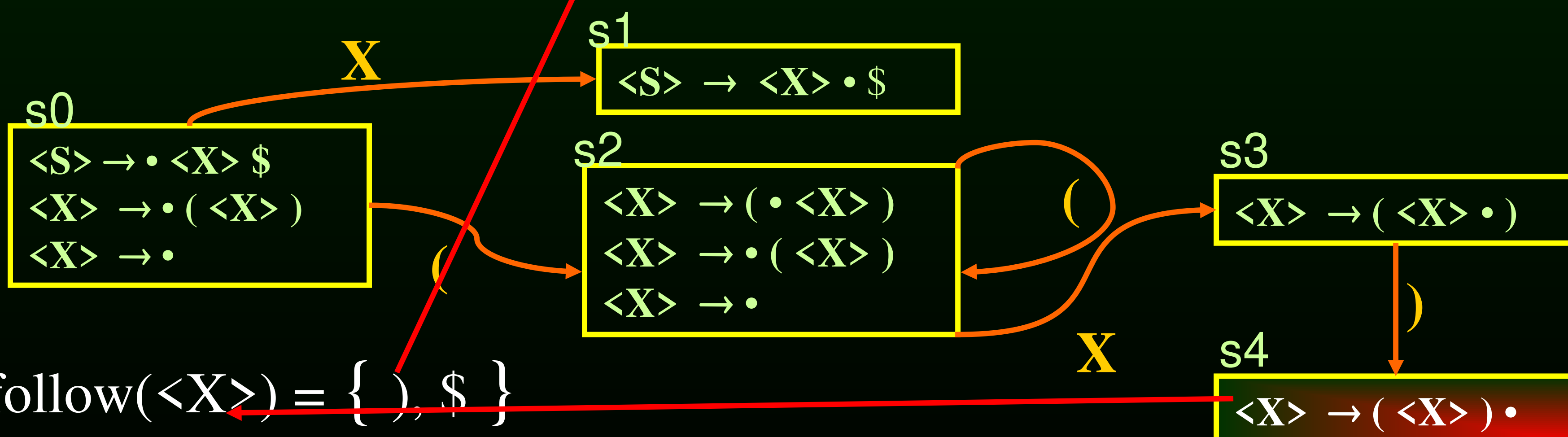


follow(<X>) = {), $ }

follow(<S>) = { $ }

Construyendo la tabla de parseo

	ACTION			Goto
State	(	)	$	X
s0	shift to s2	reduce (3)	reduce (3)	goto s1
s1	error	error	accept	
s2	shift to s2	reduce (3)	reduce(3)	goto s3
s3	error	shift to s4	error	
s4		reduce (2)		

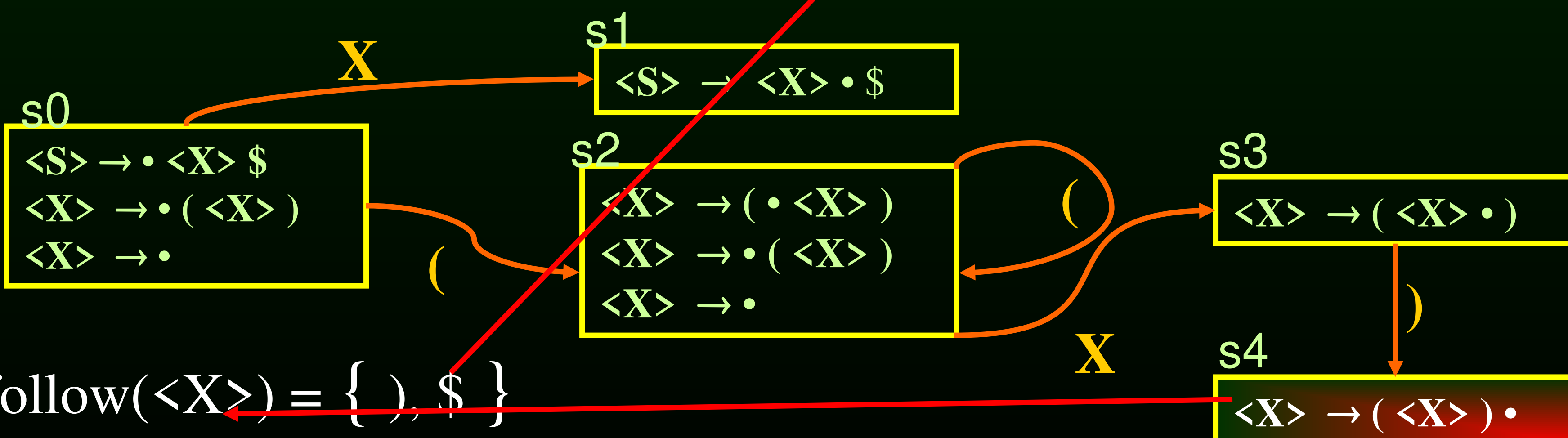


follow(<X>) = {), $ }

follow(<S>) = { $ }

Construyendo la tabla de parseo

	ACTION			Goto
State	(	)	$	X
s0	shift to s2	reduce (3)	reduce (3)	goto s1
s1	error	error	accept	
s2	shift to s2	reduce (3)	reduce(3)	goto s3
s3	error	shift to s4	error	
s4		reduce (2)	reduce (2)	

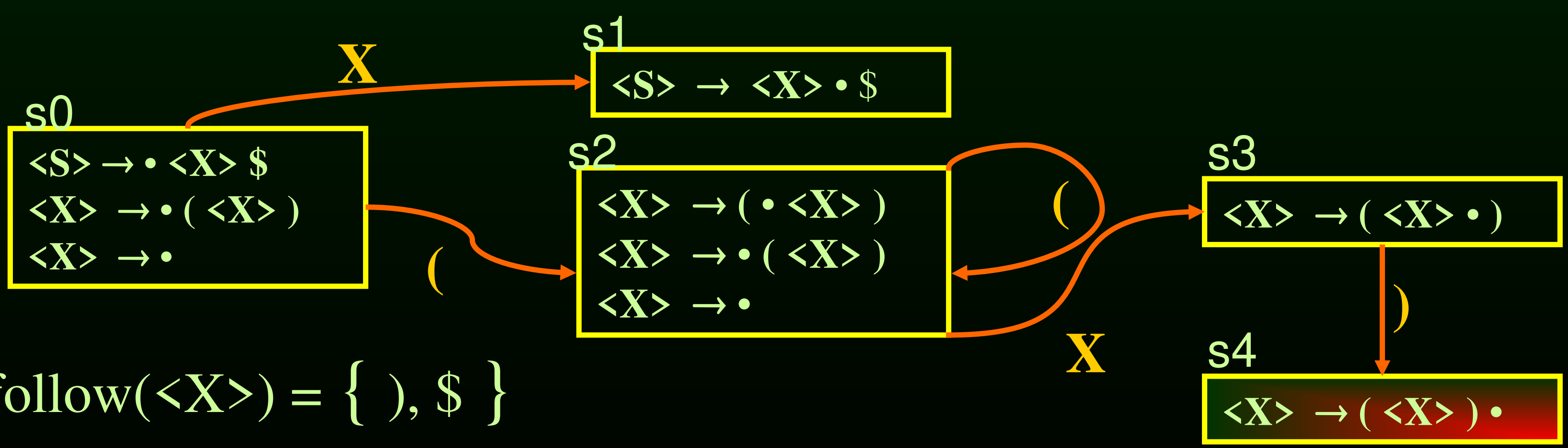


follow(<X>) = {), $ }

follow(<S>) = { $ }

Construyendo la tabla de parseo

		ACTION		Goto
State	(	)	$	X
s0	shift to s2	reduce (3)	reduce (3)	goto s1
s1	error	error	accept	
s2	shift to s2	reduce (3)	reduce(3)	goto s3
s3	error	shift to s4	error	
s4		reduce (2)	reduce (2)	

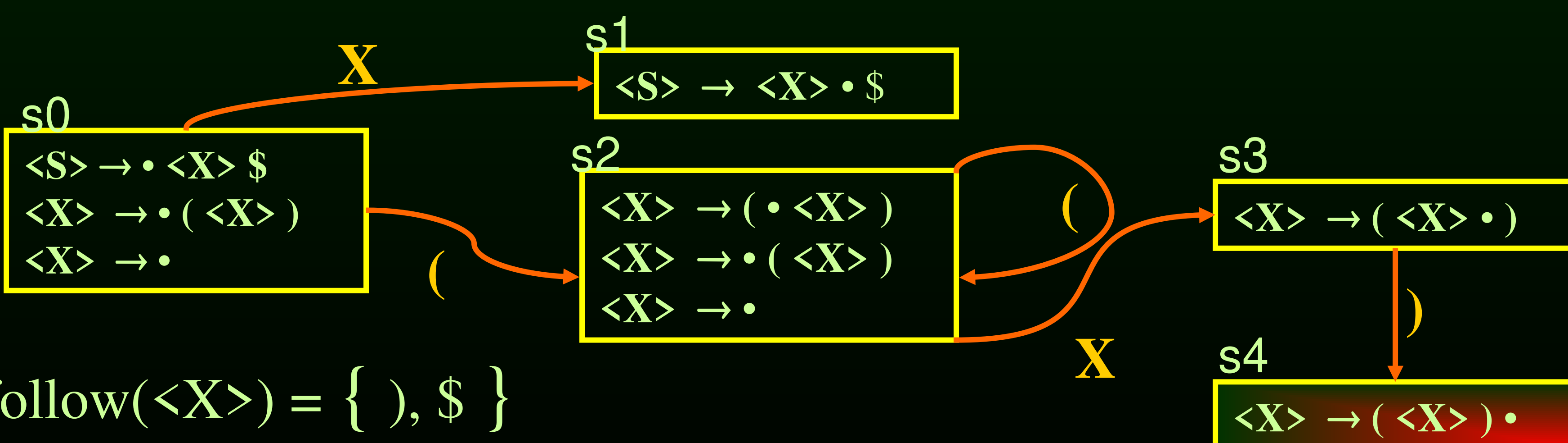


follow(<X>) = {), $ }

follow(<S>) = { $ }

Construyendo la tabla de parseo

	ACTION			Goto
State	(	)	$	X
s0	shift to s2	reduce (3)	reduce (3)	goto s1
s1	error	error	accept	
s2	shift to s2	reduce (3)	reduce(3)	goto s3
s3	error	shift to s4	error	
s4	error	reduce (2)	reduce (2)	

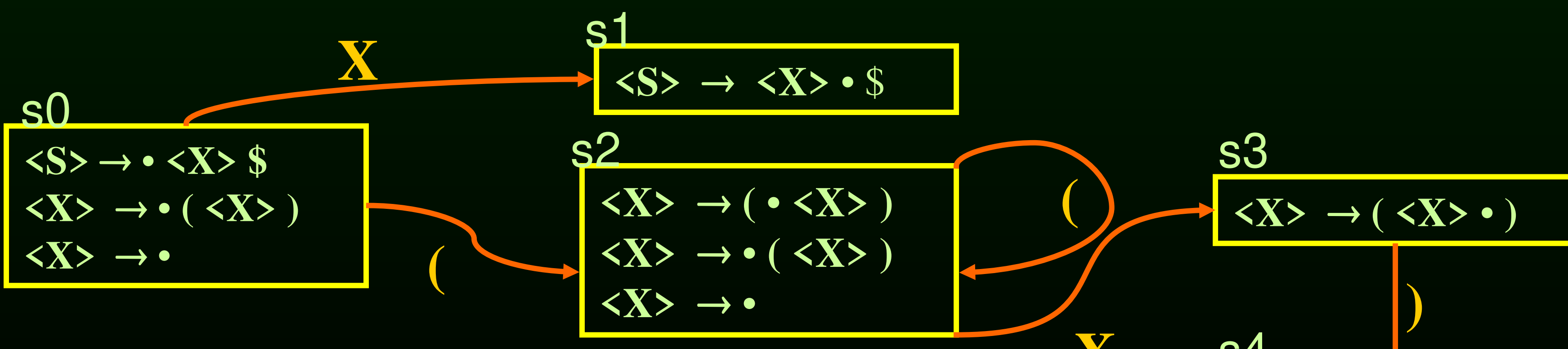


follow(<X>) = {), $ }

follow(<S>) = { $ }

Construyendo la tabla de parseo

State	ACTION (	)	$	Goto X
s0	shift to s2	reduce (3)	reduce (3)	goto s1
s1	error	error	accept	
s2	shift to s2	reduce (3)	reduce(3)	goto s3
s3	error	shift to s4	error	
s4	error	reduce (2)	reduce (2)	



follow(<X>) = {), $ }

follow(<S>) = { $ }

Tablas de parseo LR(0) y SLR(1)

SLR(1)

	ACTION			Goto
State	(	)	$	X
s0	shift to s2	reduce (3)	reduce(3)	goto s1
s1	error	error	accept	
s2	shift to s2	reduce (3)	reduce(3)	goto s3
s3	error	shift to s4	error	
s4	error	reduce (2)	reduce (2)	

LR(0)

	ACTION			Goto
State	(	)	$	X
s0	shift to s2/reduce (3)	reduce (3)	reduce(3)	goto s1
s1	error	error	accept	
s2	shift to s2/reduce (3)	reduce (3)	reduce(3)	goto s3
s3	error	shift to s4	error	
s4	reduce(2)	reduce(2)	reduce(2)	

Parser Engine SLR(1)

- Usamos los ítems
  - Sólo vemos la posición actual
  - Sin conocimiento del input futuro
  - SLR(0) ???
- En la reducción
  - Vemos el siguiente input (usando el conjunto follow)
  - Por lo tanto SLR(1)

Resumen

- Resumen de la construcción de un parser LR(0)
- Limitaciones de los lenguajes LR(0)
- Construcción de un parser engine SLR(1)
- Limitaciones de los lenguajes SLR(1)
- Construcción de un parser engine LR(1)
- Construcción de un parser engine LALR(1)

Gramáticas SLR(1) vs LR(1)

- SLR(1) mantiene la posición actual además de alguna información acerca del siguiente caracter de entrada
- LR(1) mantiene información completa acerca del siguiente caracter de entrada
- Examinemos una gramática que es LR(1) pero no SLR(1)

Ejemplo Expandido

- La gramática

  <S> → <X> $

  <X> → (<X>)

  <X> → ( )

- El lenguaje:
  - Cero o más paréntesis abiertos seguidos de un # igual de paréntesis cerrados
- Cambiemos el lenguaje a:
  - Cero o más paréntesis abiertos seguidos de un # igual de paréntesis cerrados
  - o un solo paréntesis abierto

Ejemplo Expandido

Ejemplos:

“((((()))))” “()” “(“ “”

- Cambiemos el lenguaje a:
  - Cero o más paréntesis abiertos seguidos de un # igual de paréntesis cerrados
  - o un solo paréntesis abierto

Gramática del Ejemplo Expandido

<S> → <X> $ (1)

<X> → <Y> (2)

<X> → ((3)

<Y> → (<Y>) (4)

<Y> → ε (5)

- Cambiemos el lenguaje a:
  - Cero o más paréntesis abiertos seguidos de un # igual de paréntesis cerrados
  - o un solo paréntesis abierto

Items del Ejemplo Expandido

$\langle S \rangle \rightarrow \langle X \rangle \; \$$

$\langle X \rangle \rightarrow \langle Y \rangle$

$\langle X \rangle \rightarrow ($

$\langle Y \rangle \rightarrow (\; \langle Y \rangle \;)$

$\langle Y \rangle \rightarrow \varepsilon$

- Items

  $\langle S \rangle \rightarrow \bullet \langle X \rangle \; \$$

  $\langle S \rangle \rightarrow \langle X \rangle \bullet \$$

  $\langle X \rangle \rightarrow \bullet \langle Y \rangle$

  $\langle X \rangle \rightarrow \langle Y \rangle \bullet$

  $\langle X \rangle \rightarrow \bullet ($

  $\langle X \rangle \rightarrow (\bullet$

  $\langle Y \rangle \rightarrow \bullet (\; \langle Y \rangle \;)$

  $\langle Y \rangle \rightarrow (\bullet \langle Y \rangle \;)$

  $\langle Y \rangle \rightarrow (\; \langle Y \rangle \bullet)$

  $\langle Y \rangle \rightarrow (\; \langle Y \rangle \;) \bullet$

  $\langle Y \rangle \rightarrow \bullet$

DFA del Ejemplo Expandido

<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε

DFA del Ejemplo Expandido

<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε

s0

<S> → • <X> $
<X> → • <Y>
<X> → • (
<Y> → • (<Y>)
<Y> → •

DFA del Ejemplo Expandido

$\langle S \rangle \rightarrow \langle X \rangle \ \$$

$\langle X \rangle \rightarrow \langle Y \rangle$

$\langle X \rangle \rightarrow ($

$\langle Y \rangle \rightarrow (\ \langle Y \rangle \ )$

$\langle Y \rangle \rightarrow \varepsilon$

s0

$\langle S \rangle \rightarrow \bullet \ \langle X \rangle \ \$$
$\langle X \rangle \rightarrow \bullet \ \langle Y \rangle$
$\langle X \rangle \rightarrow \bullet \ ($
$\langle Y \rangle \rightarrow \bullet \ (\ \langle Y \rangle \ )$
$\langle Y \rangle \rightarrow \bullet$

(

DFA del Ejemplo Expandido

<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε

s0

<S> → • <X> $
<X> → • <Y>
<X> → • (
<Y> → • (<Y>)
<Y> → •

(

s1

<X> → (•
<Y> → (• <Y>)
<Y> → • (<Y>)
<Y> → •

DFA del Ejemplo Expandido

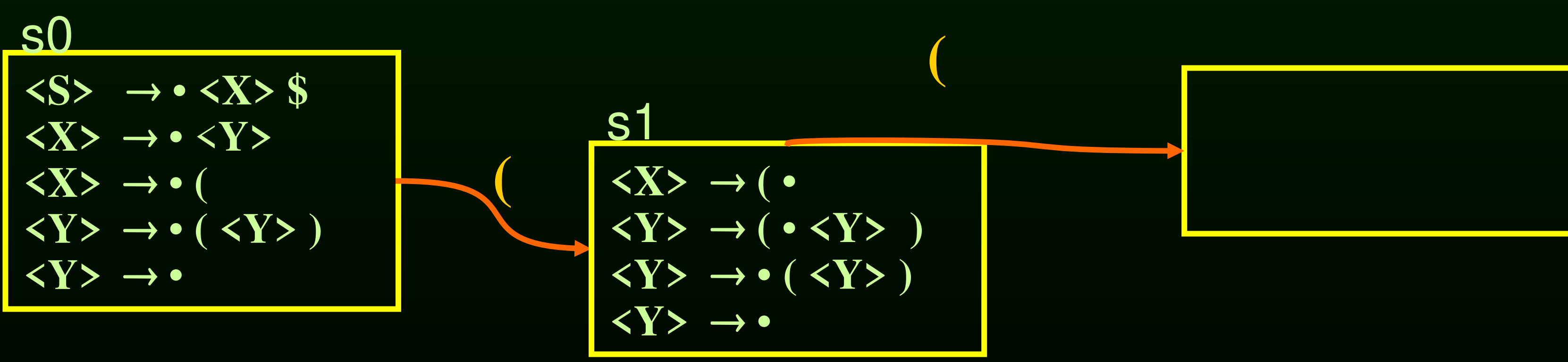
<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε



DFA del Ejemplo Expandido

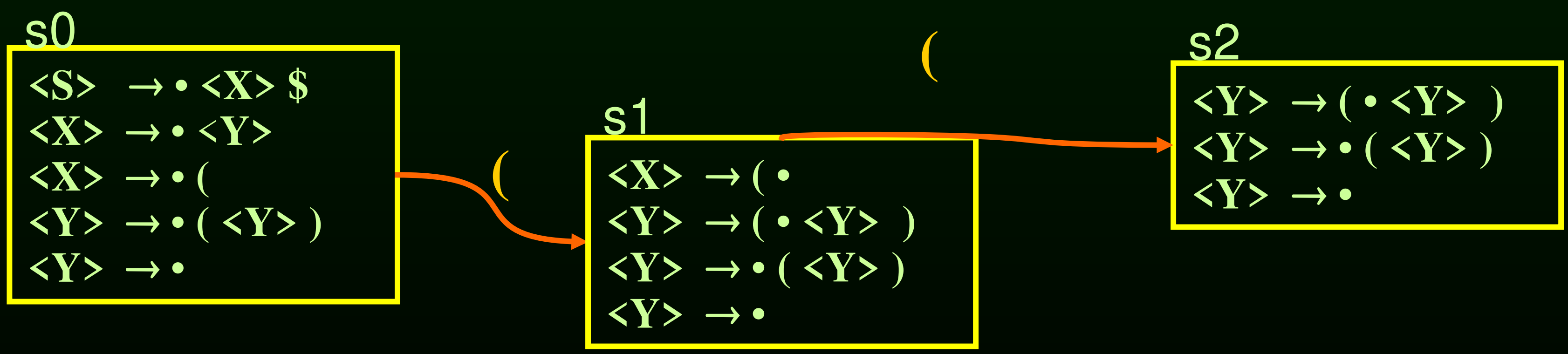
$\langle S \rangle \rightarrow \langle X \rangle \ \$$

$\langle X \rangle \rightarrow \langle Y \rangle$

$\langle X \rangle \rightarrow ($

$\langle Y \rangle \rightarrow (\ \langle Y \rangle \ )$

$\langle Y \rangle \rightarrow \varepsilon$



DFA del Ejemplo Expandido

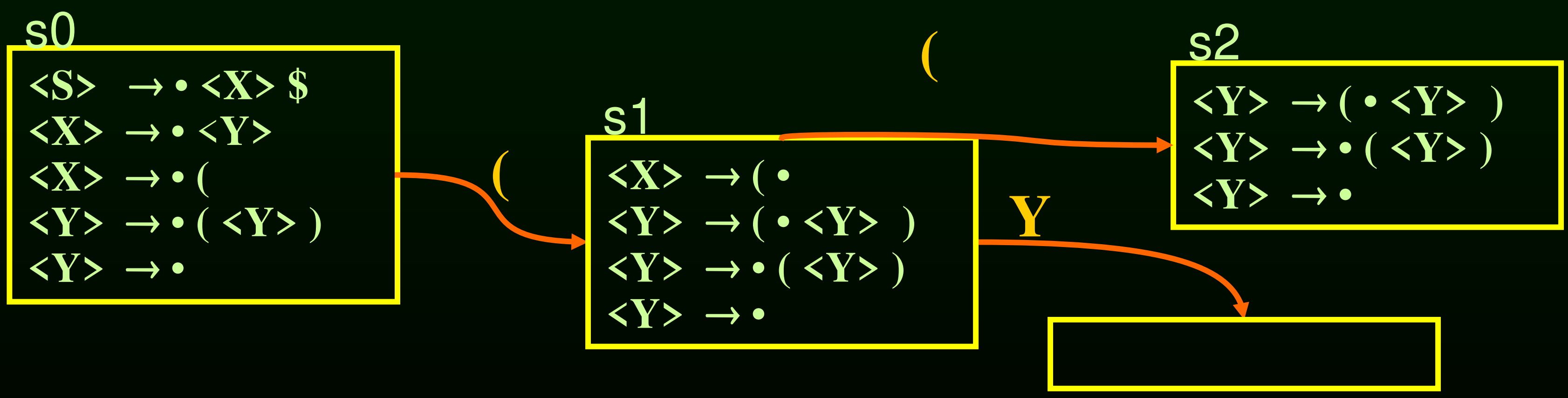
<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε



DFA del Ejemplo Expandido

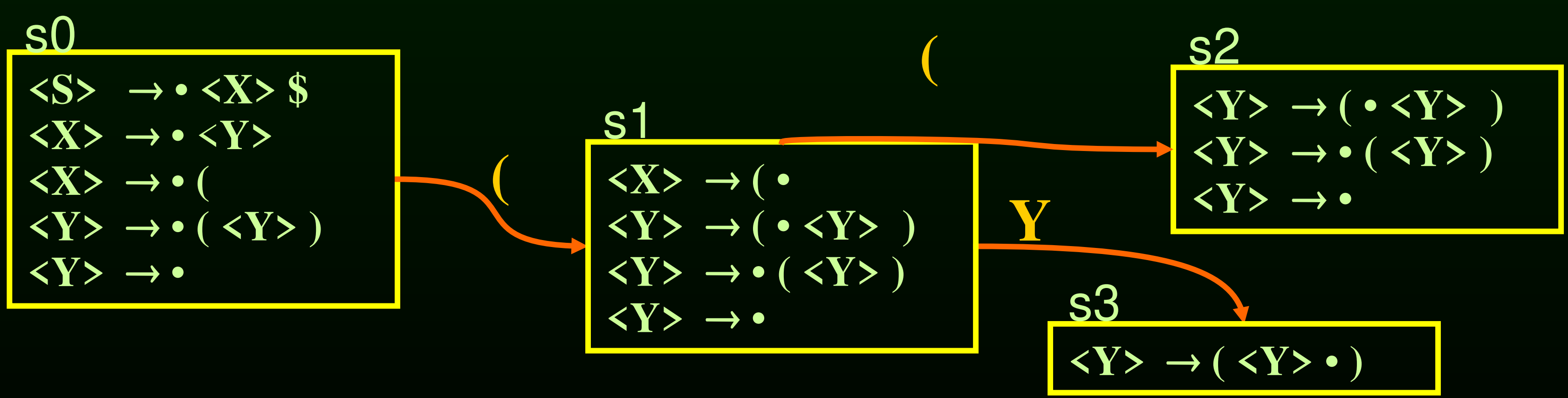
<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε



DFA del Ejemplo Expandido

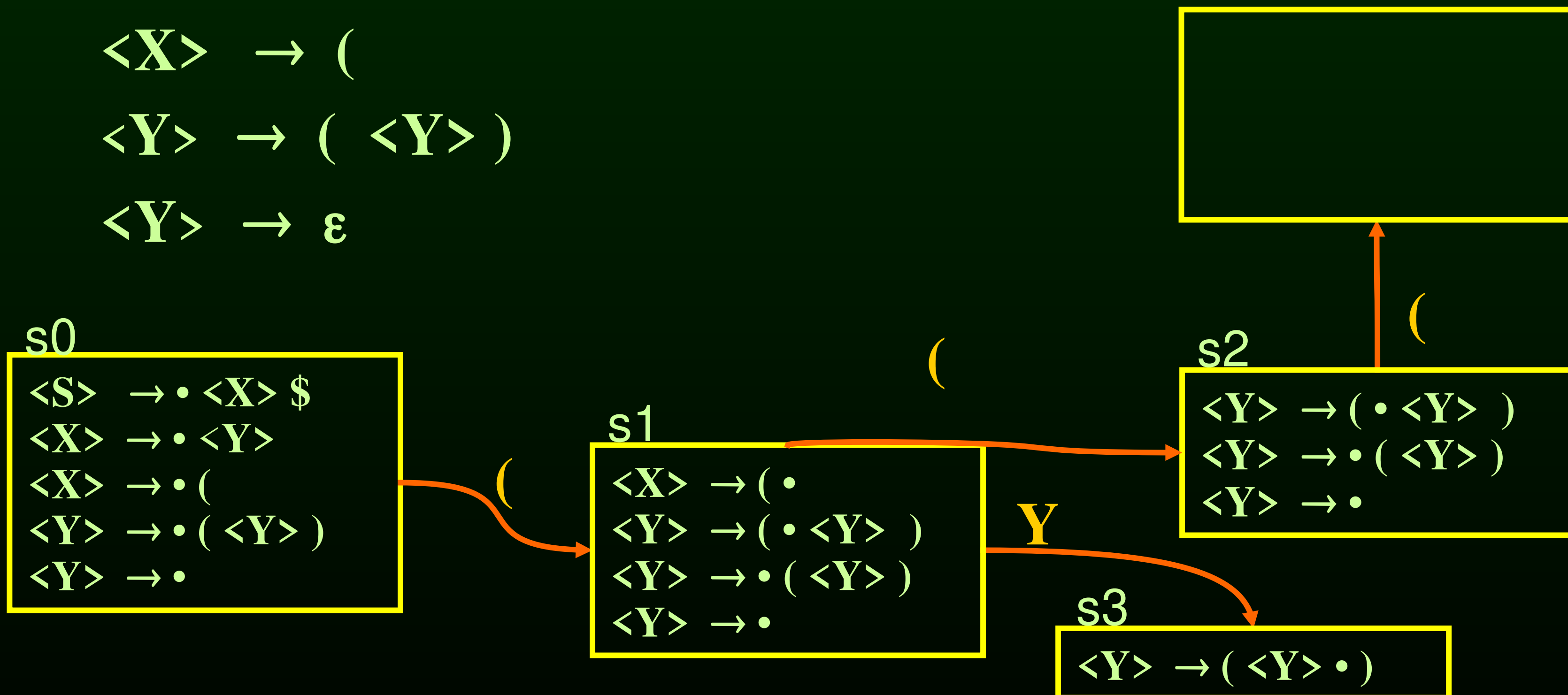
<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε



DFA del Ejemplo Expandido

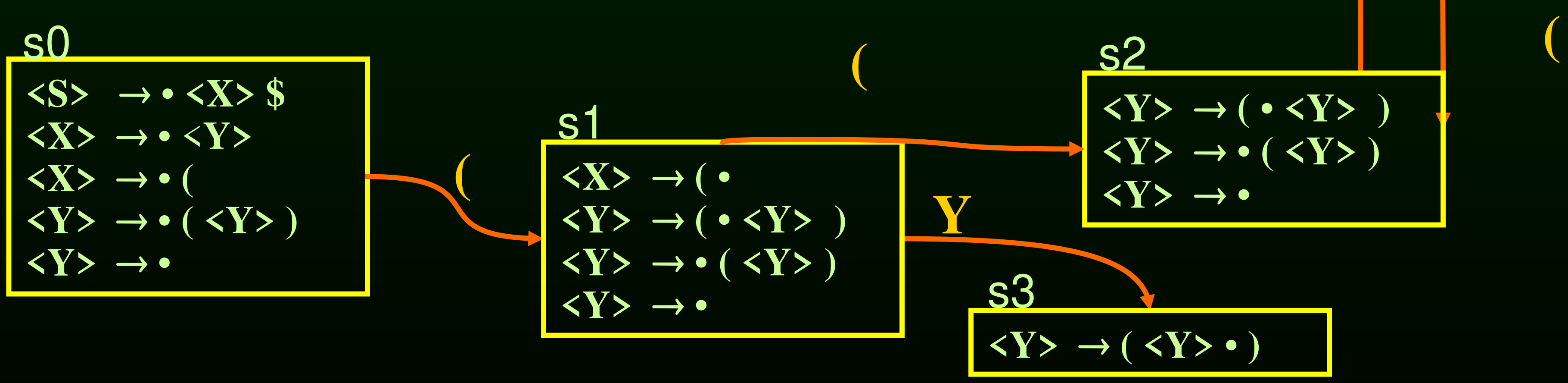
<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε



DFA del Ejemplo Expandido

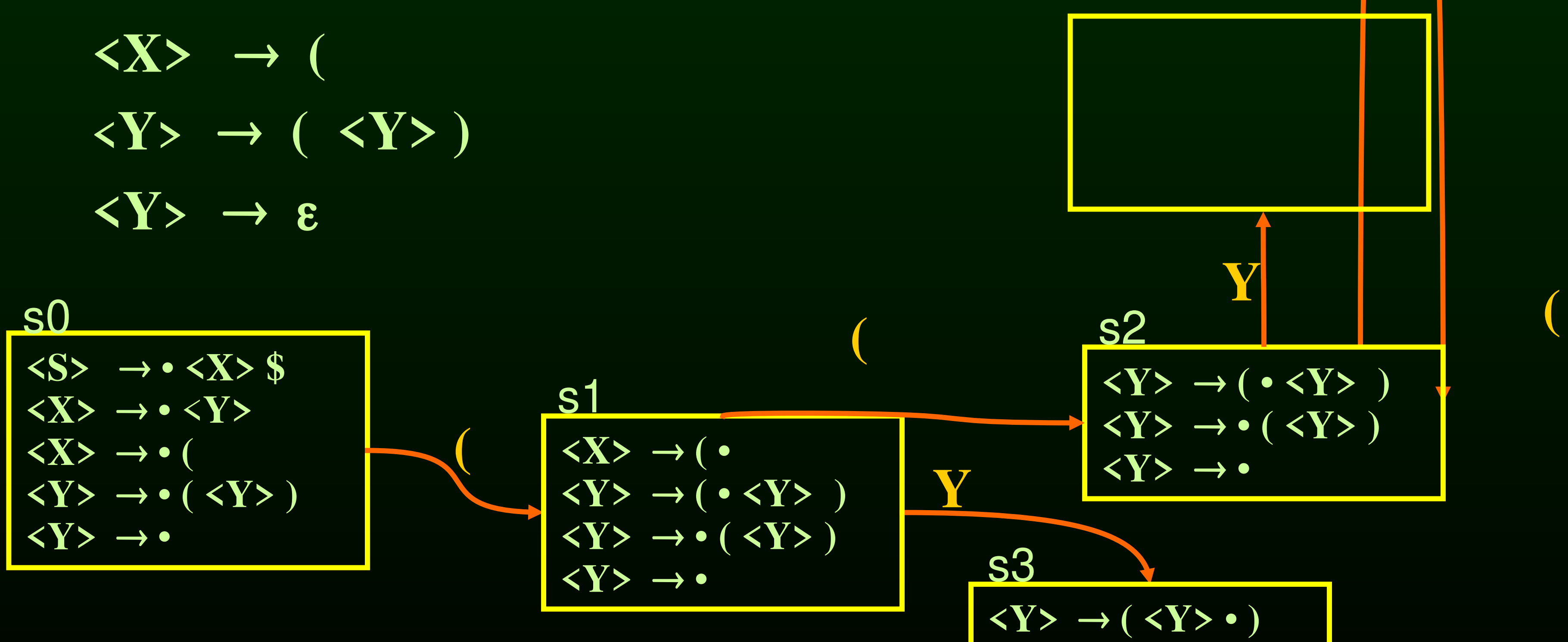
<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε



DFA del Ejemplo Expandido

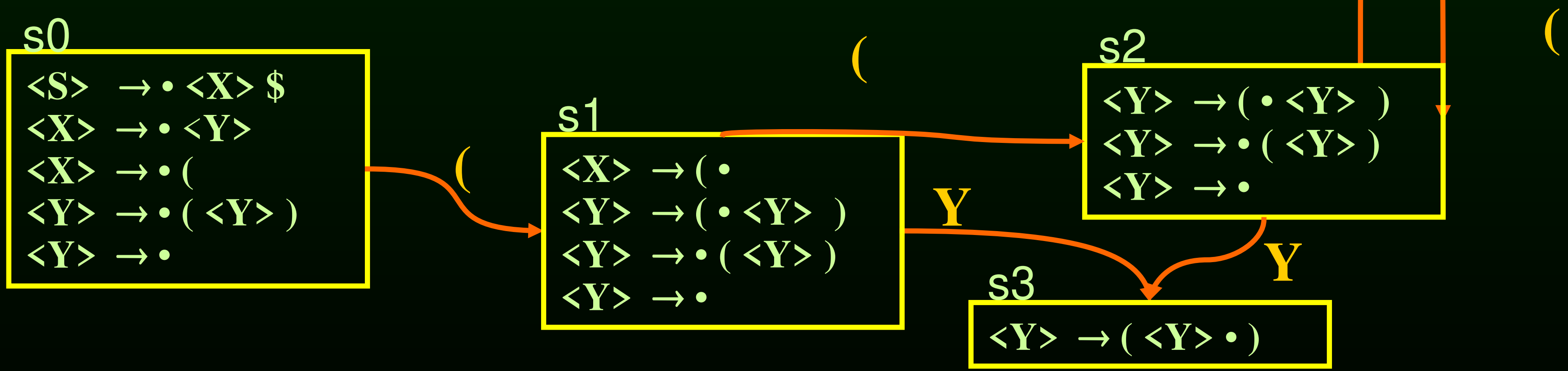
<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε



DFA del Ejemplo Expandido

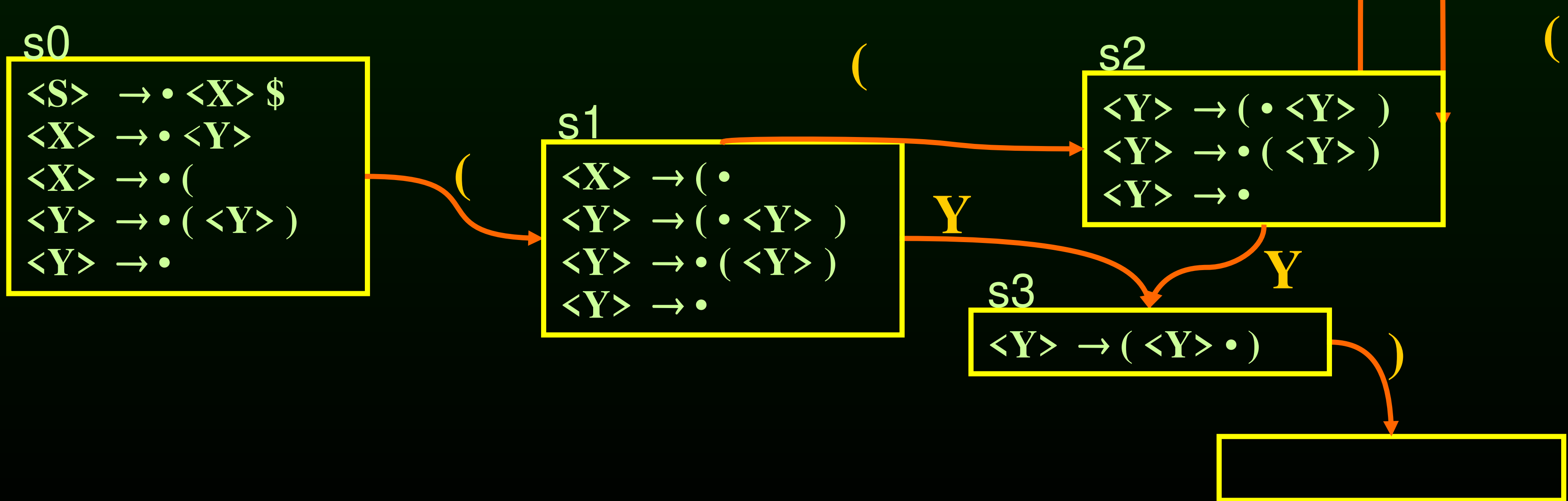
<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε



DFA del Ejemplo Expandido

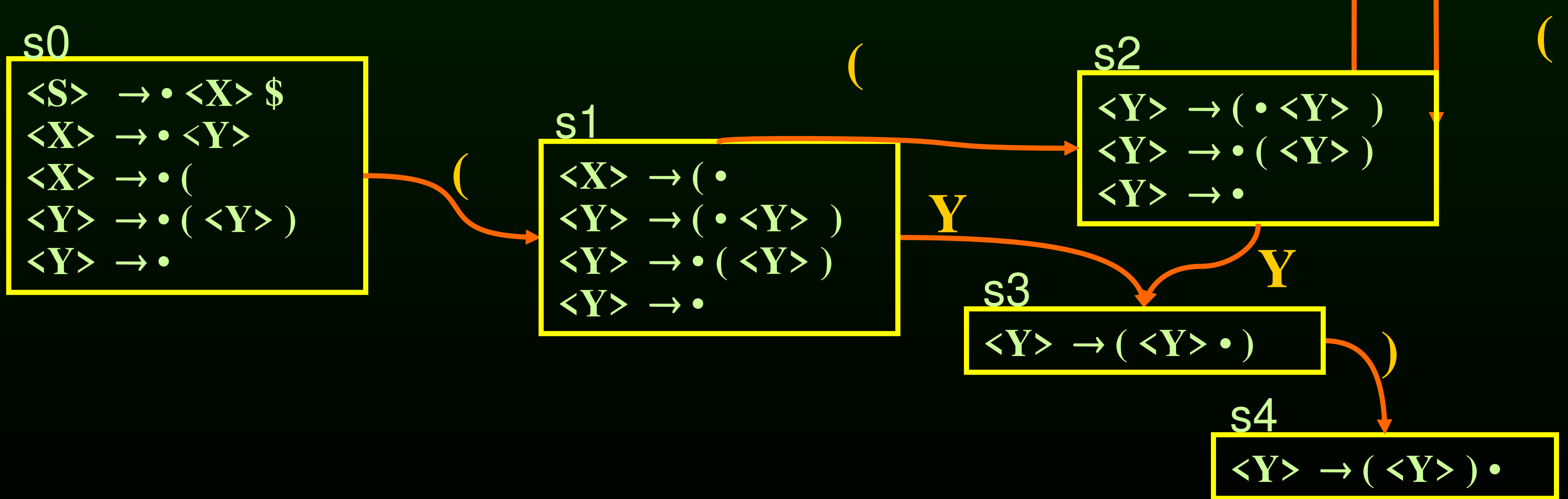
$\langle S \rangle \rightarrow \langle X \rangle\ \$$

$\langle X \rangle \rightarrow \langle Y \rangle$

$\langle X \rangle \rightarrow ($

$\langle Y \rangle \rightarrow (\ \langle Y \rangle \ )$

$\langle Y \rangle \rightarrow \varepsilon$



DFA del Ejemplo Expandido

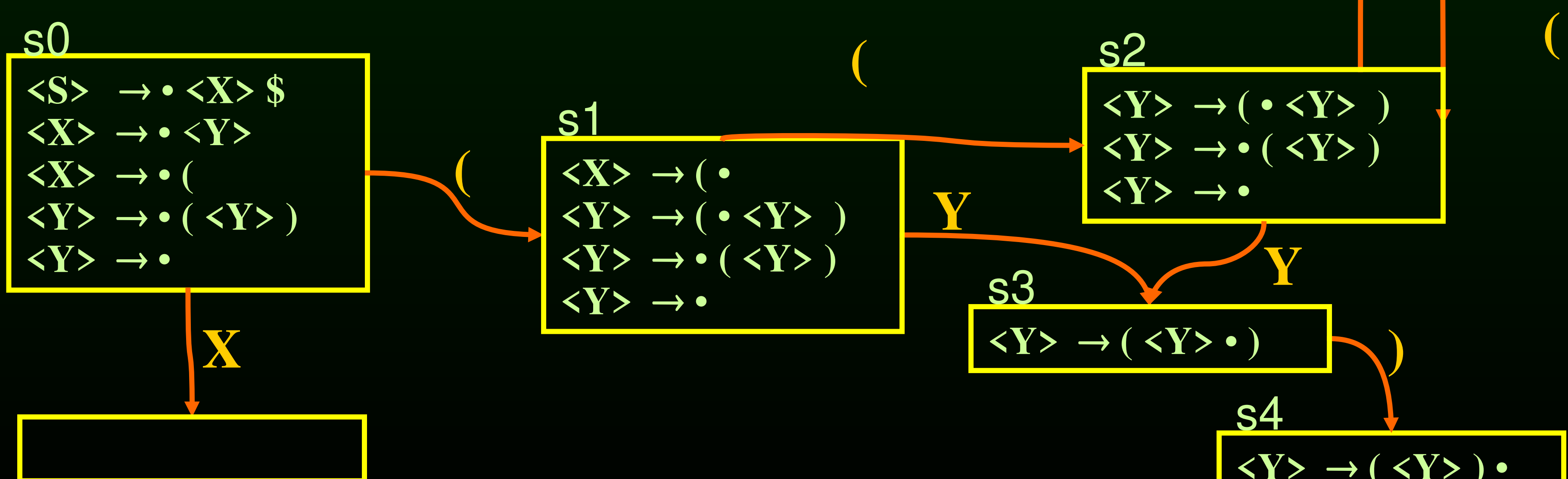
<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε



DFA del Ejemplo Expandido

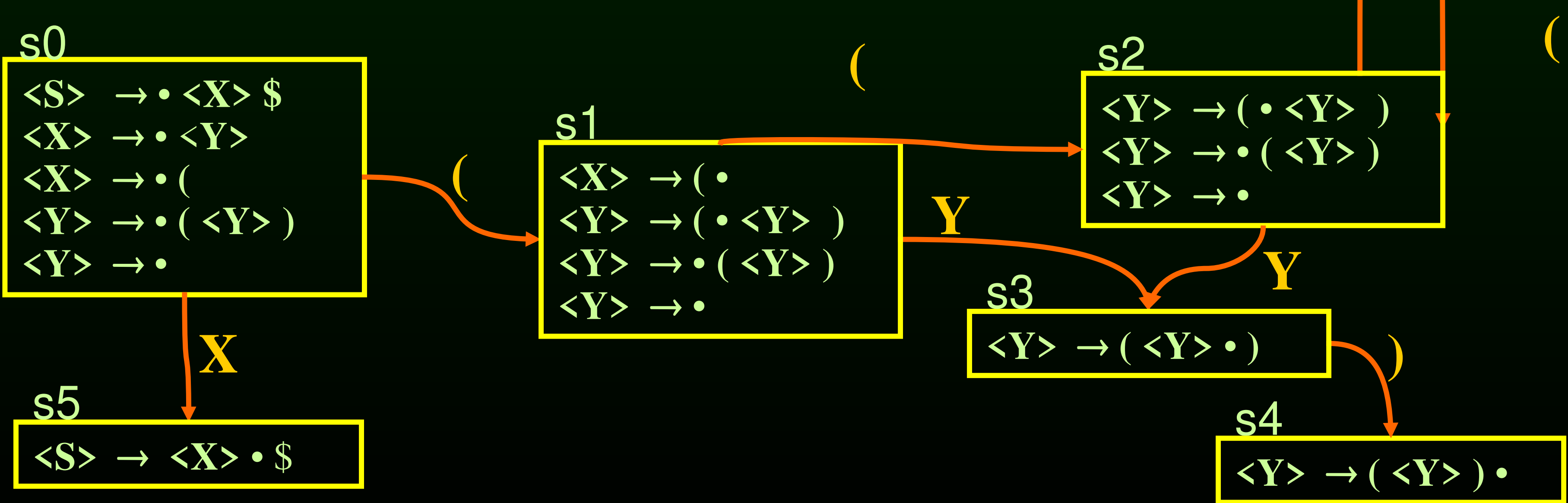
<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε



DFA del Ejemplo Expandido

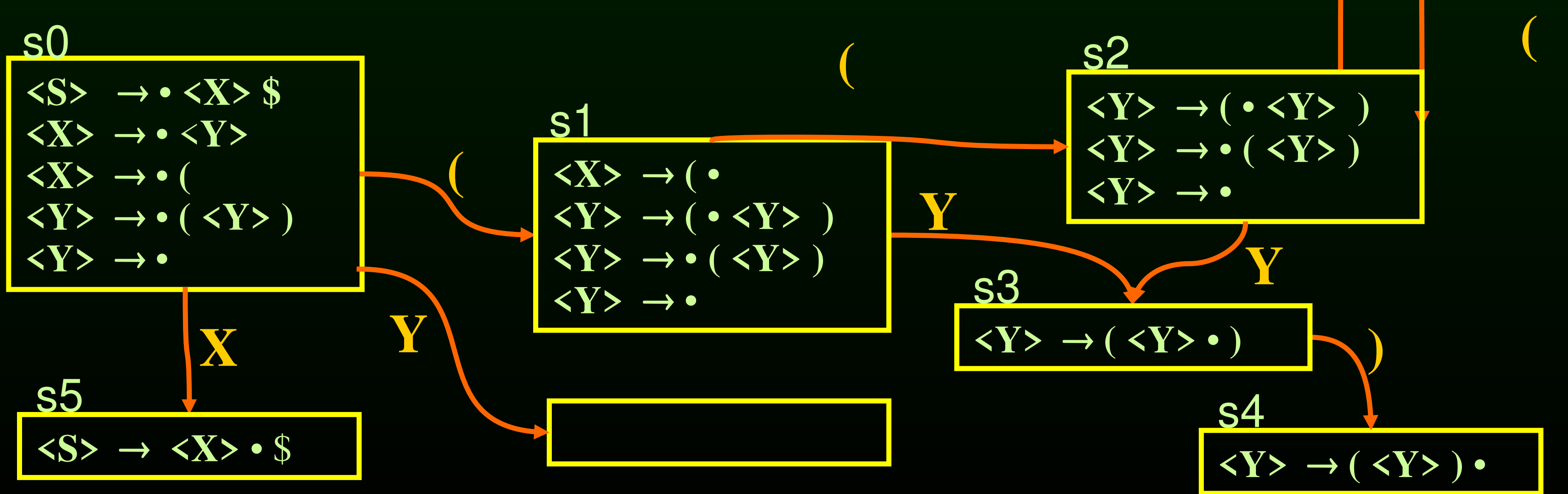
<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε



DFA del Ejemplo Expandido

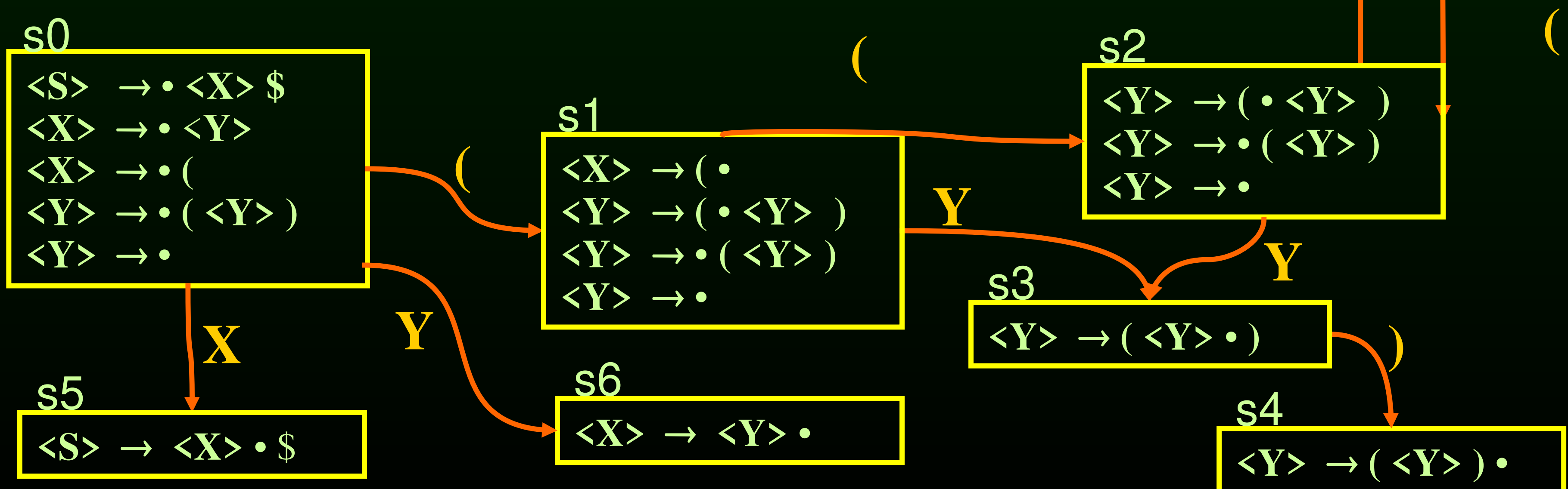
<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε



Conjuntos follow del Ejemplo Expandido

- **follow(<S>) = ???**
- **follow(<X>) = ???**
- **follow(<Y>) = ???**

- gramática

  <S> → <X> $

  <X> → <Y>

  <X> → (

  <Y> → (<Y>)

  <Y> → ε

Conjuntos follow del Ejemplo Expandido

- **follow(<S>) =** { $ }
- **follow(<X>) = ???**
- **follow(<Y>) = ???**

- gramática

  <S> → <X> $

  <X> → <Y>

  <X> → (

  <Y> → (<Y>)

  <Y> → ε

Conjuntos follow del Ejemplo Expandido

- **follow(<S>) =** { $ }
- **follow(<X>) =** { $ }
- **follow(<Y>) = ???**

- gramática

  <S> → <X> $

  <X> → <Y>

  <X> → (

  <Y> → (<Y>)

  <Y> → ε

Conjuntos follow del Ejemplo Expandido

- **follow(<S>) =** { $ }
- **follow(<X>) =** { $ }
- **follow(<Y>) =** {), $ }

- gramática

  <S> → <X> $

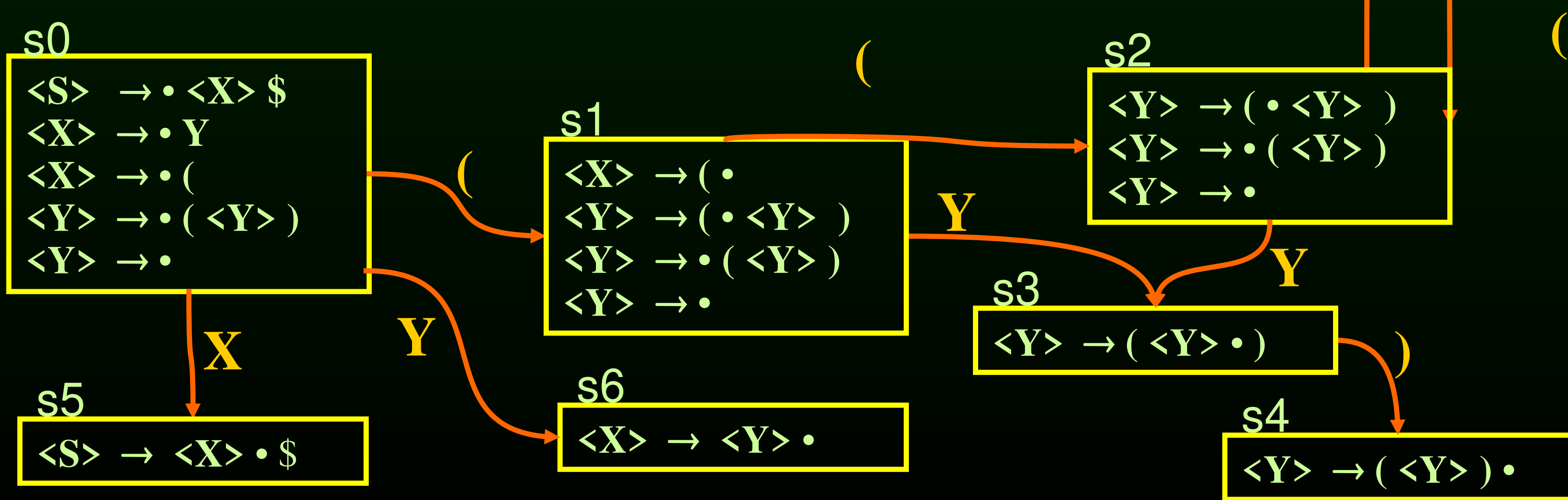
  <X> → <Y>

  <X> → (

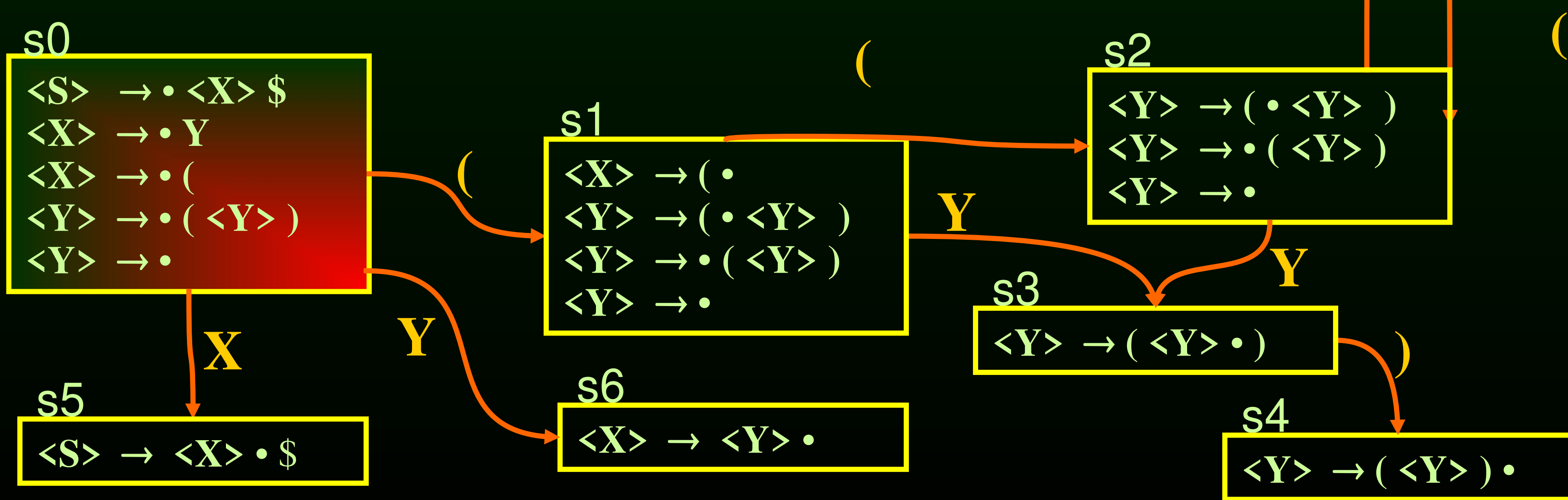
  <Y> → (<Y>)

  <Y> → ε

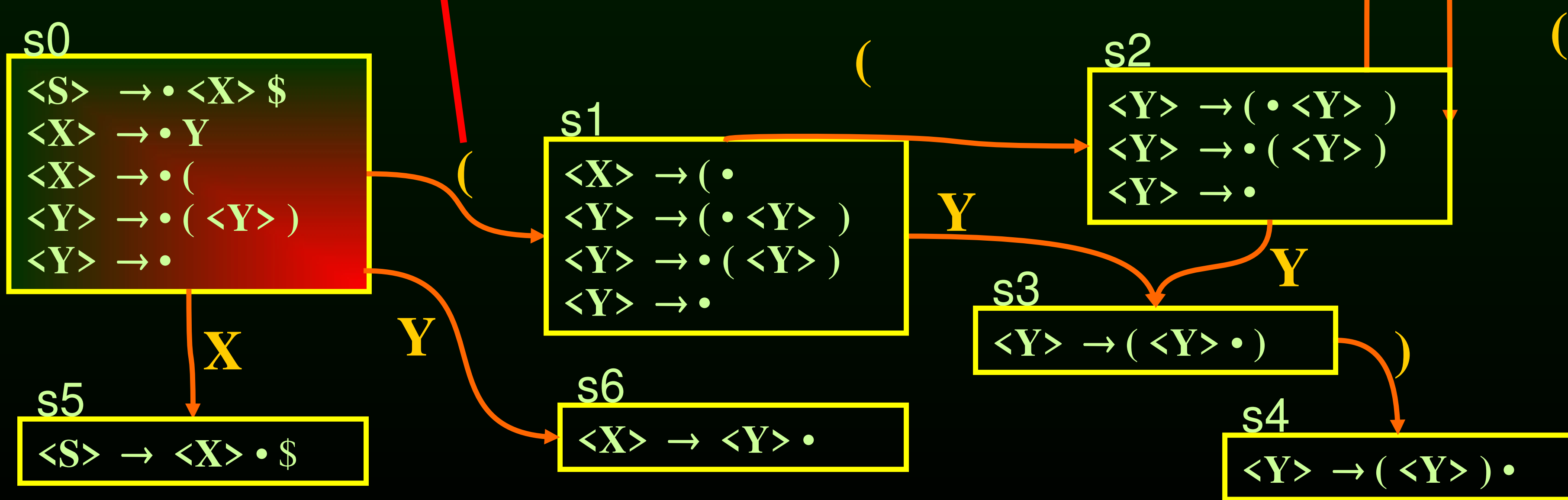
	ACTION			Goto	
State	(	)	$	X	Y
s0					
s1					
s2					
s3					
s4					
s5					
s6					



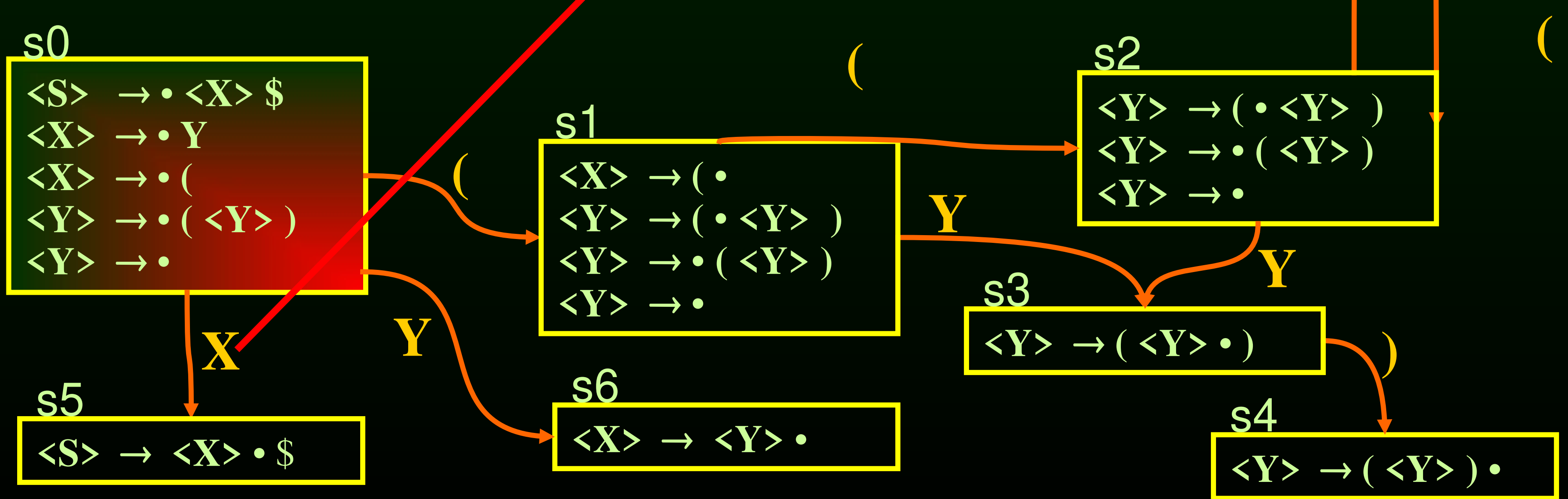
	ACTION			Goto	
State	(	)	$	X	Y
s0					
s1					
s2					
s3					
s4					
s5					
s6					



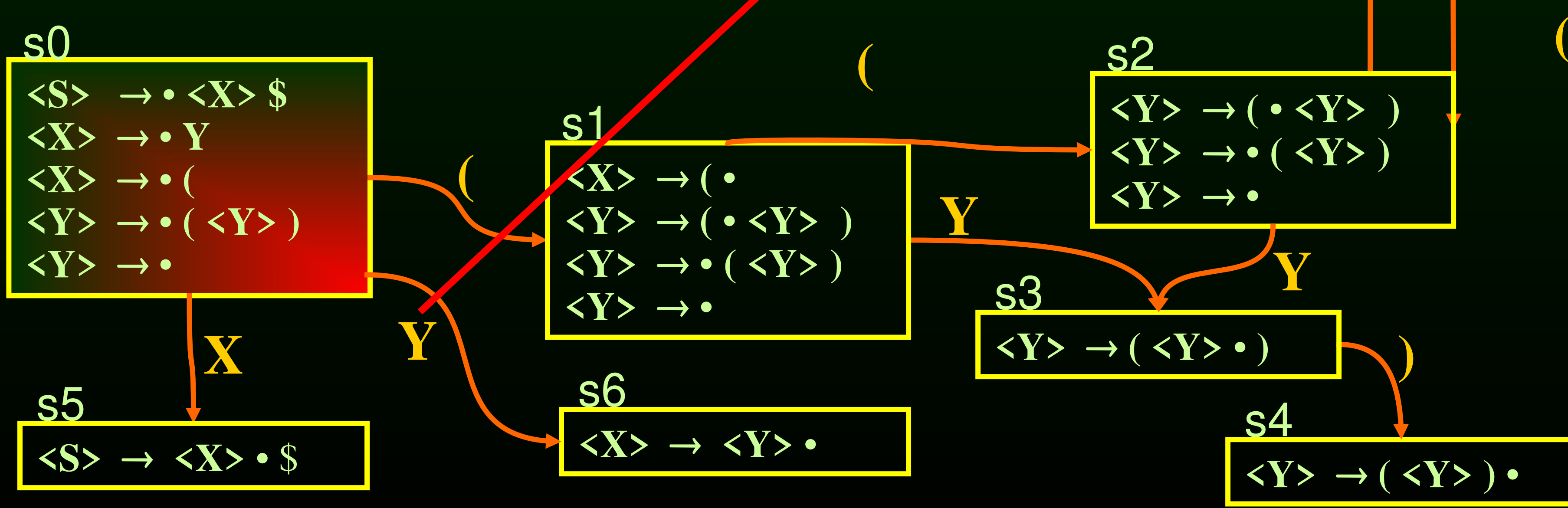
	ACTION			Goto	
State	(	)	$	X	Y
s0	shift to s1				
s1					
s2					
s3					
s4					
s5					
s6					



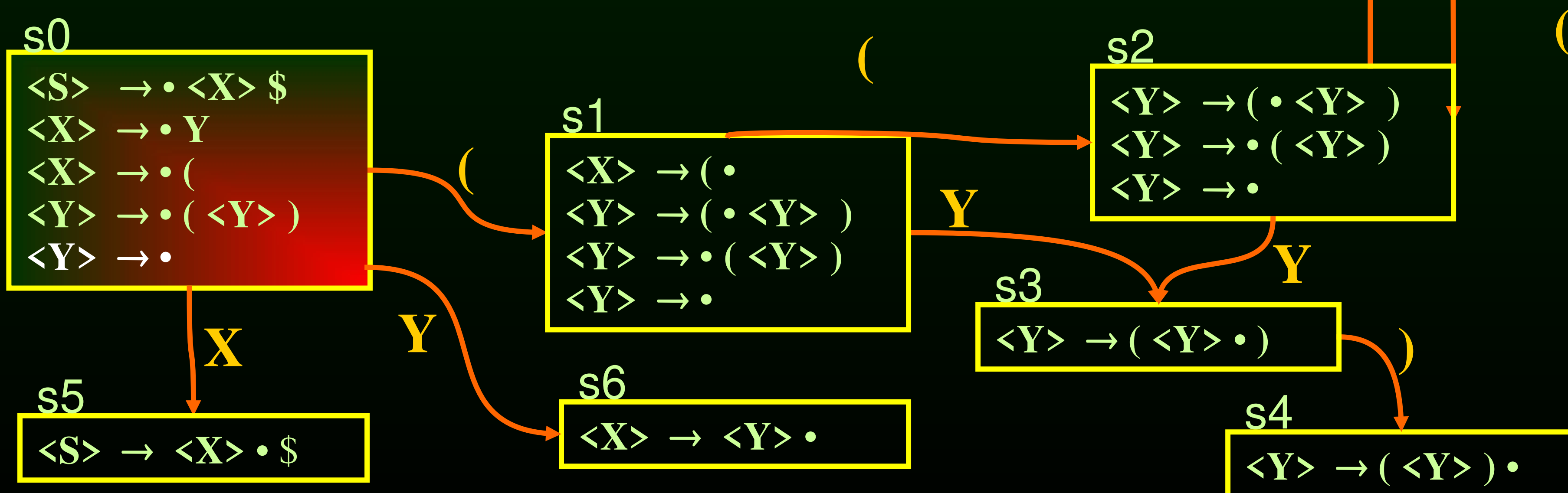
	ACTION			Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1			goto s5	
s1					
s2					
s3					
s4					
s5					
s6					



	ACTION			Goto	
State	(	)	$	X	Y
s0	shift to s1			goto s5	goto s6
s1					
s2					
s3					
s4					
s5					
s6					

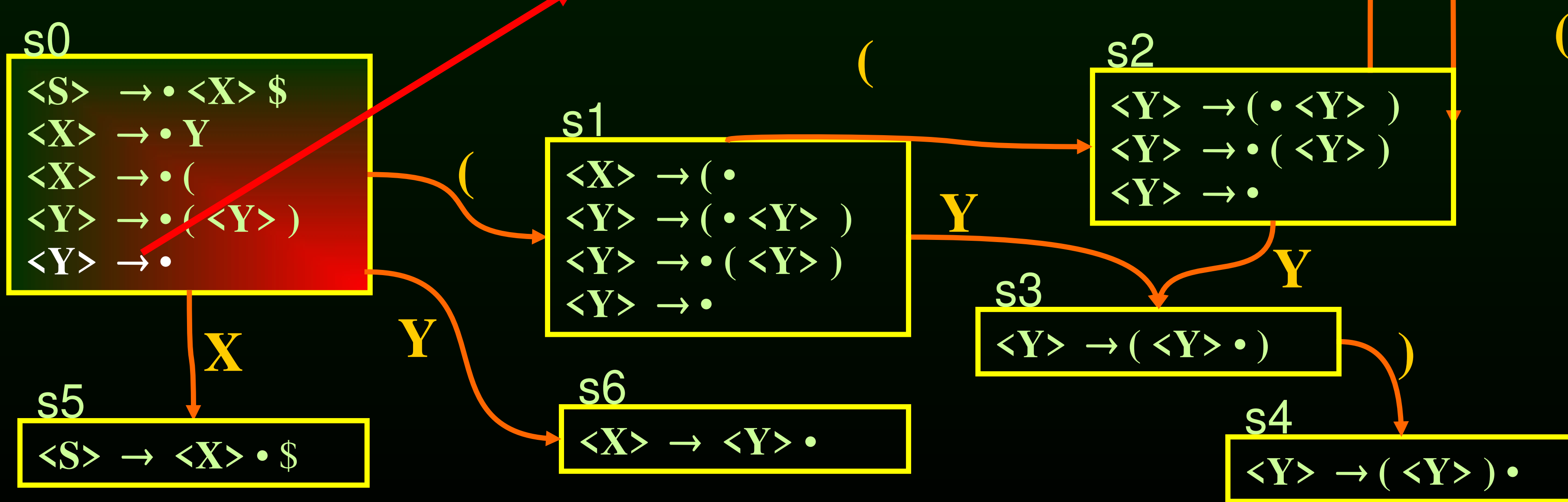


	ACTION			Goto	
State	(	)	$	X	Y
s0	shift to s1			goto s5	goto s6
s1					
s2					
s3					
s4					
s5					
s6					



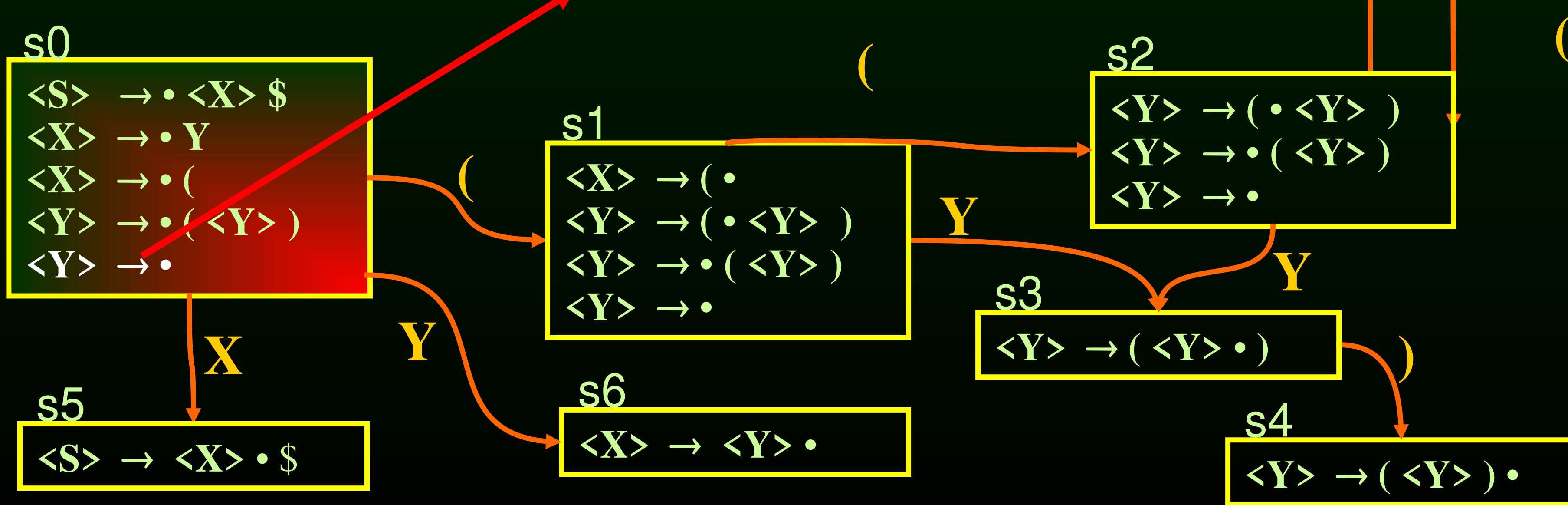
	ACTION			Goto	
State	(	)	$	X	Y
s0	shift to s1			goto s5	goto s6
s1					
s2					
s3					
s4					
s5					
s6					

Follow(<Y>) = {), $ }



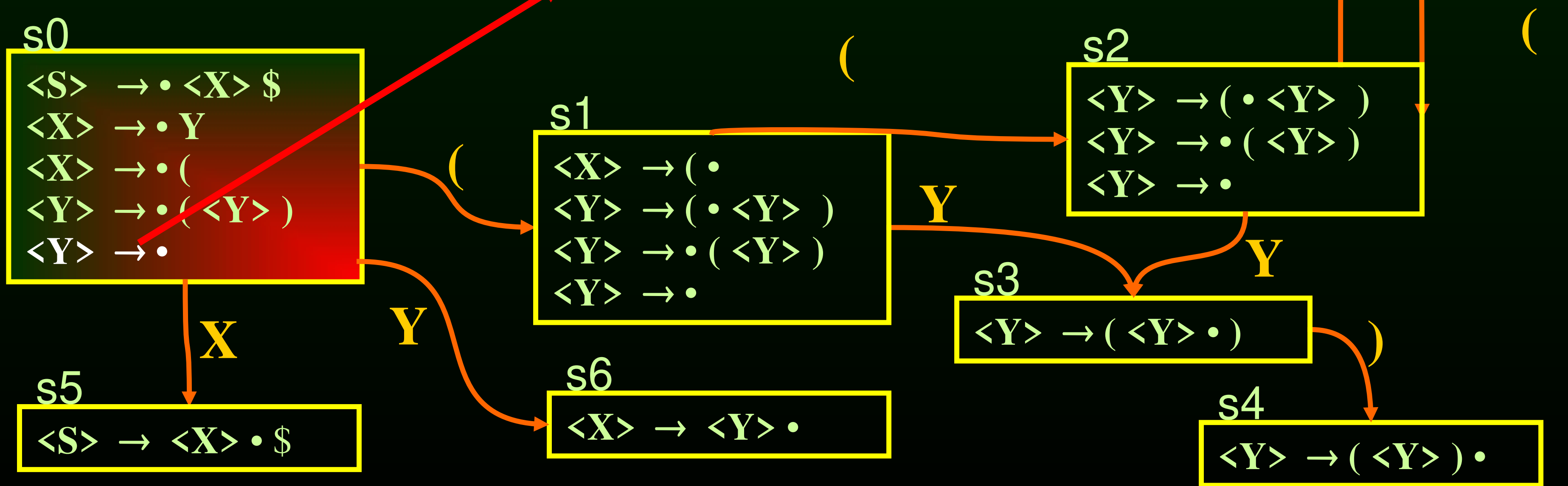
	ACTION			Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	reduce (5)		goto s5	goto s6
s1					
s2					
s3					
s4					
s5					
s6					

Follow(<Y>) = {), $ }

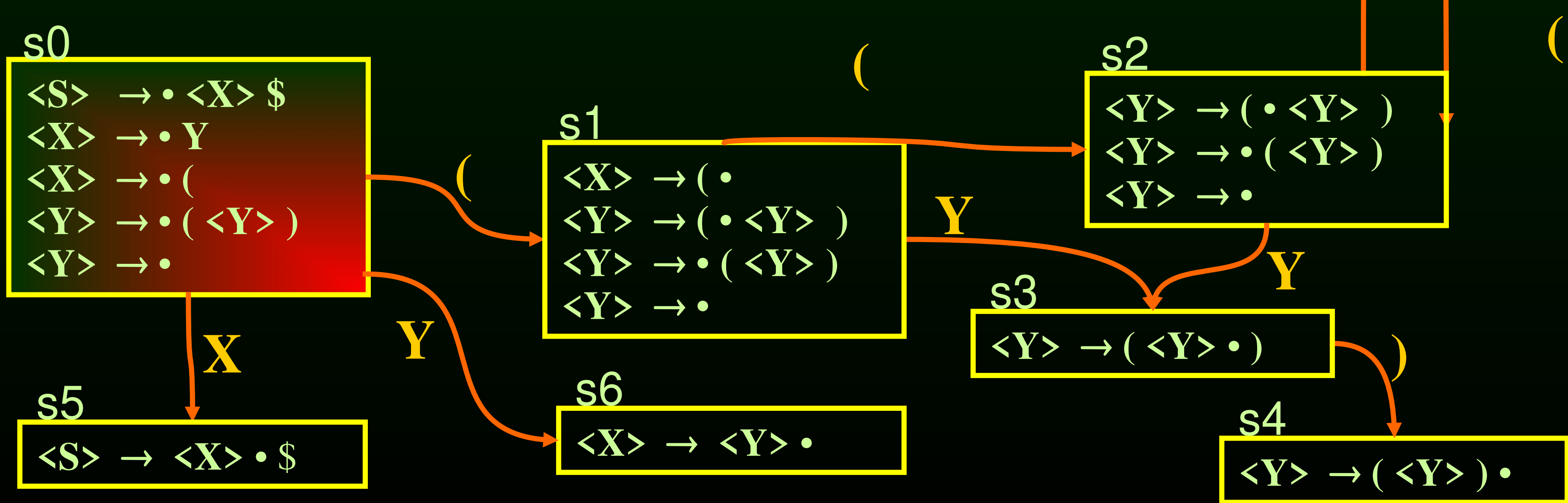


State	(	ACTION)	$	Goto X	Y
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1					
s2					
s3					
s4					
s5					
s6					

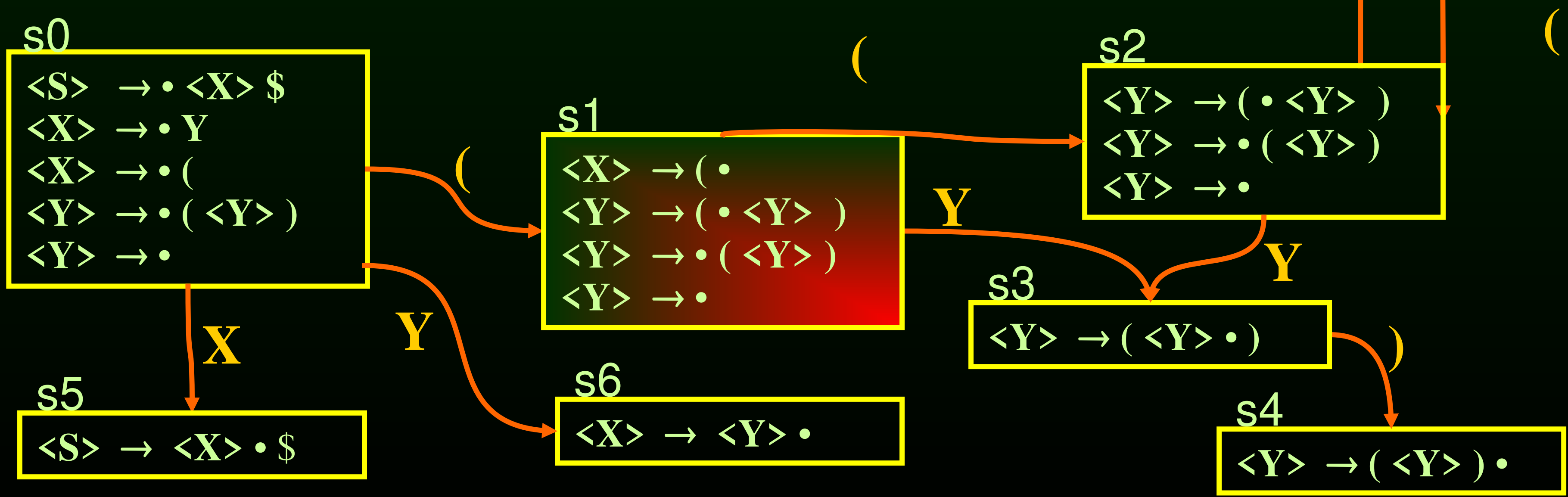
$$\text{Follow}(\langle Y\rangle) = \{\ ),\ \$\ \}$$



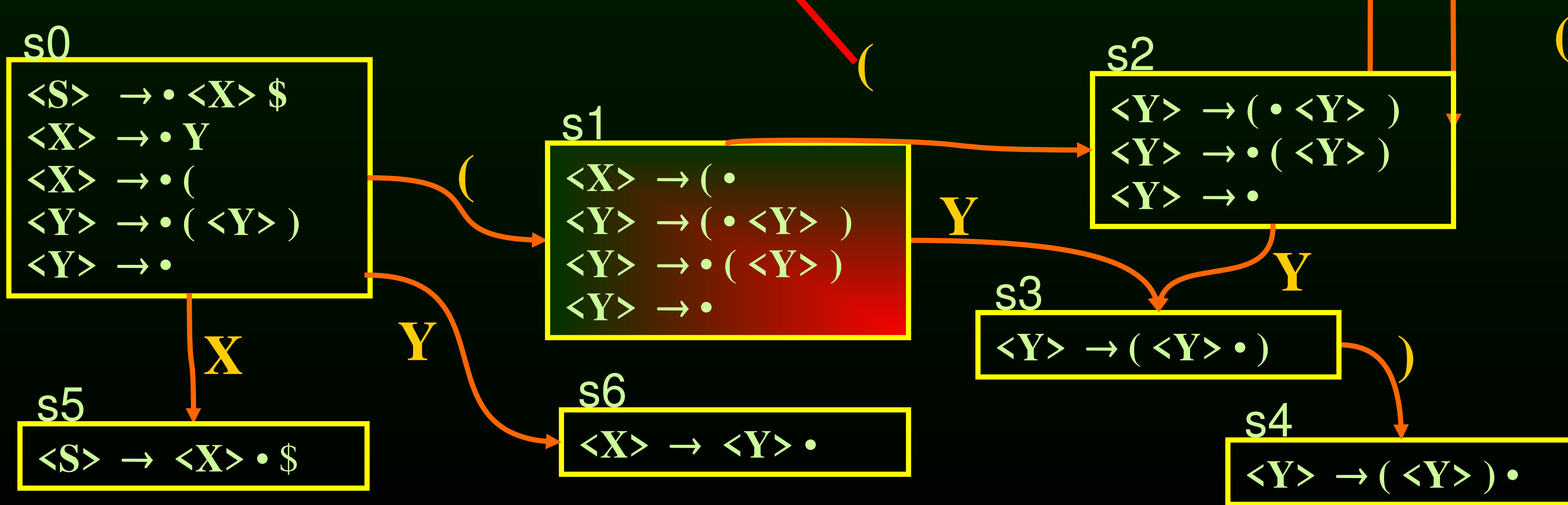
	ACTION			Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1					
s2					
s3					
s4					
s5					
s6					



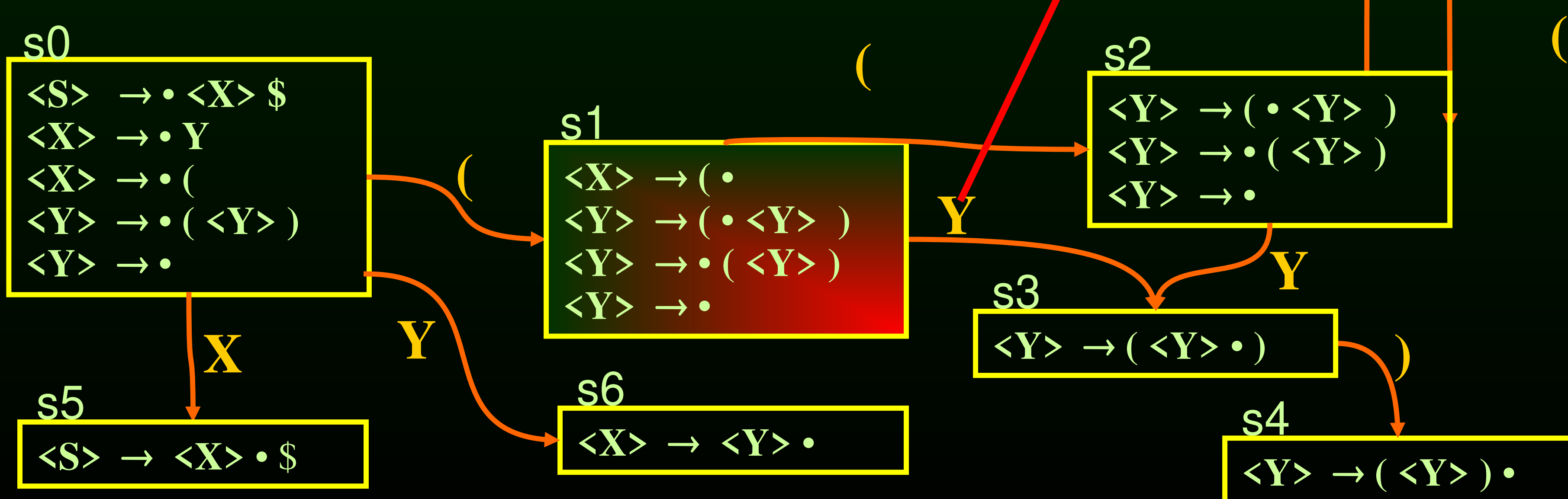
State	ACTION (	)	$	Goto X	Y
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1					
s2					
s3					
s4					
s5					
s6					



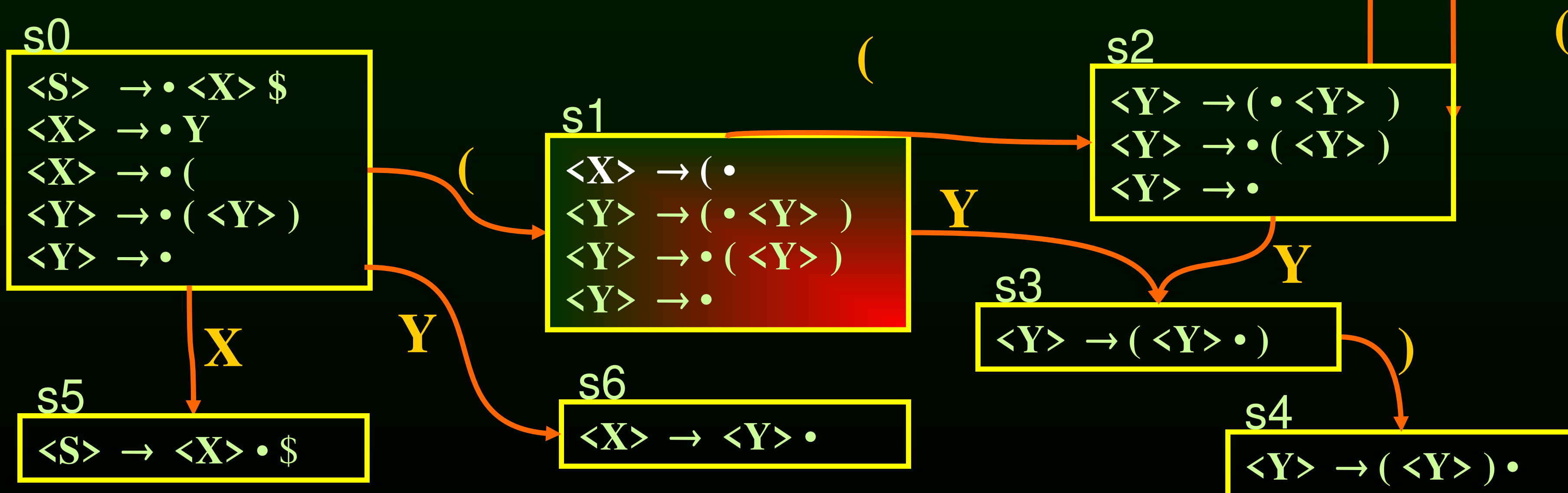
	ACTION			Goto	
State	(	)	$	X	Y
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2				
s2					
s3					
s4					
s5					
s6					



	ACTION			Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2				goto s3
s2					
s3					
s4					
s5					
s6					

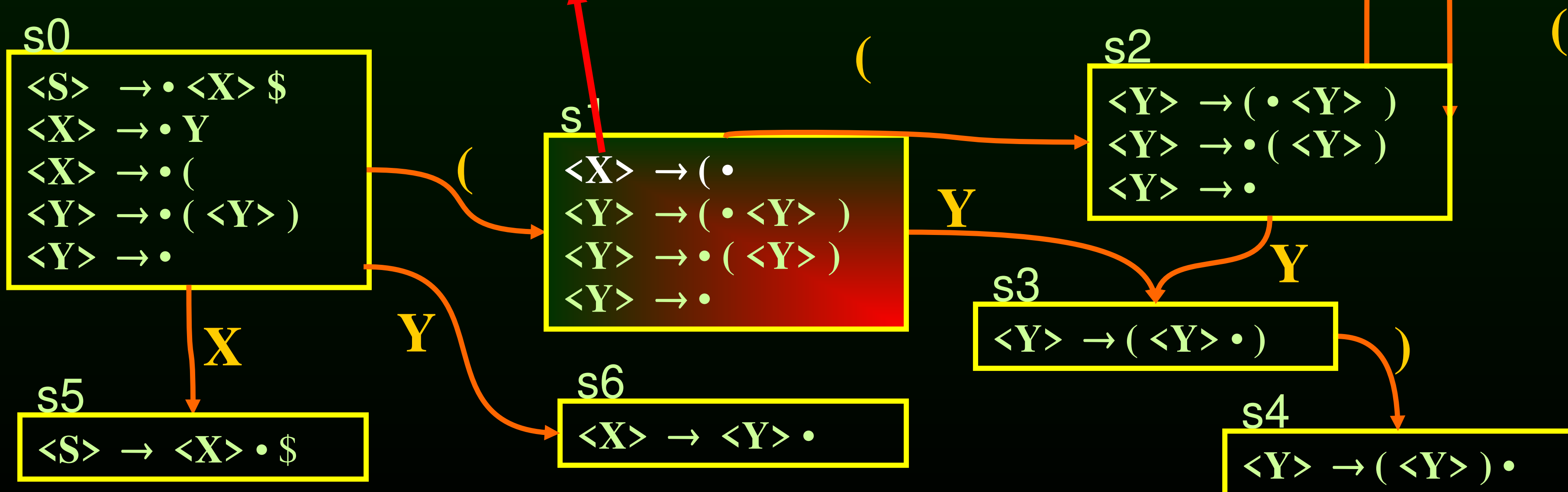


	ACTION			Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2				goto s3
s2					
s3					
s4					
s5					
s6					



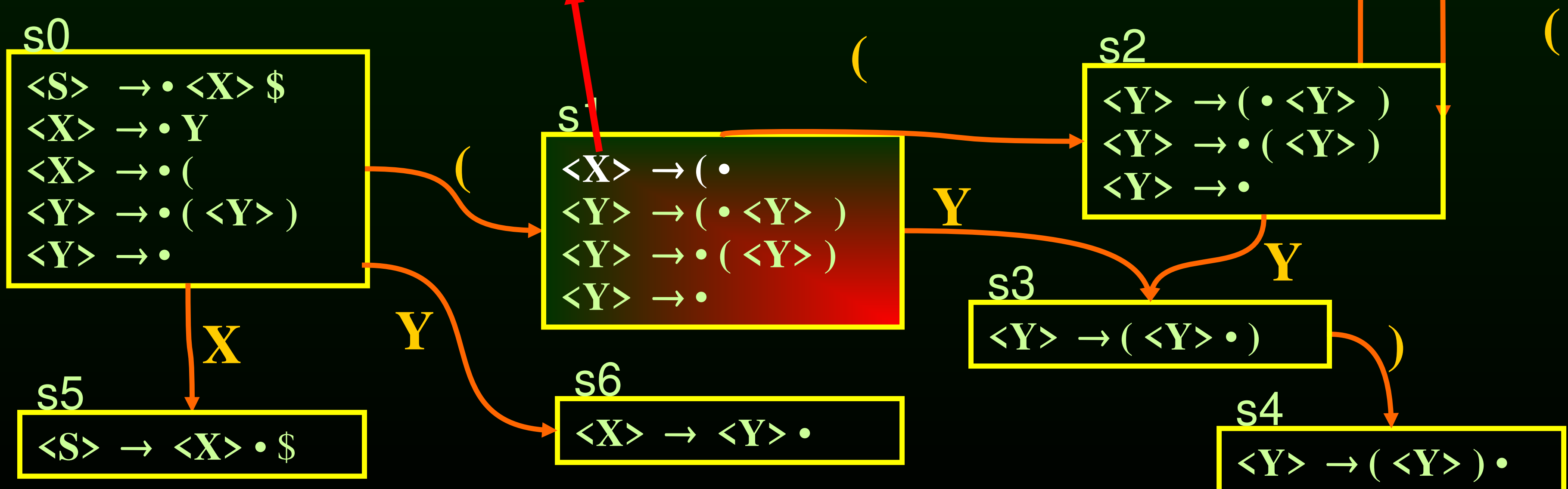
State	(	ACTION)	$	Goto X	Y
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2				goto s3
s2					
s3					
s4					
s5					
s6					

$$\text{Follow}(\langle X\rangle) = \{\ \$\ \}$$

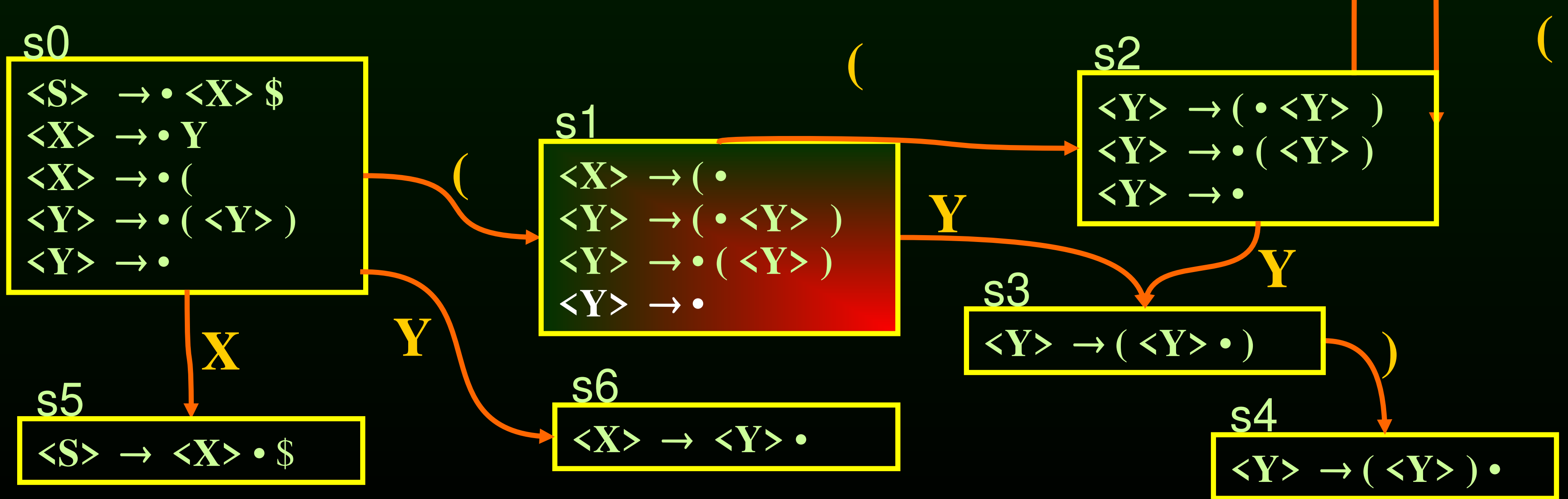


	ACTION			Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2		reduce (3)		goto s3
s2					
s3					
s4					
s5					
s6					

Follow(<X>) = { $ }

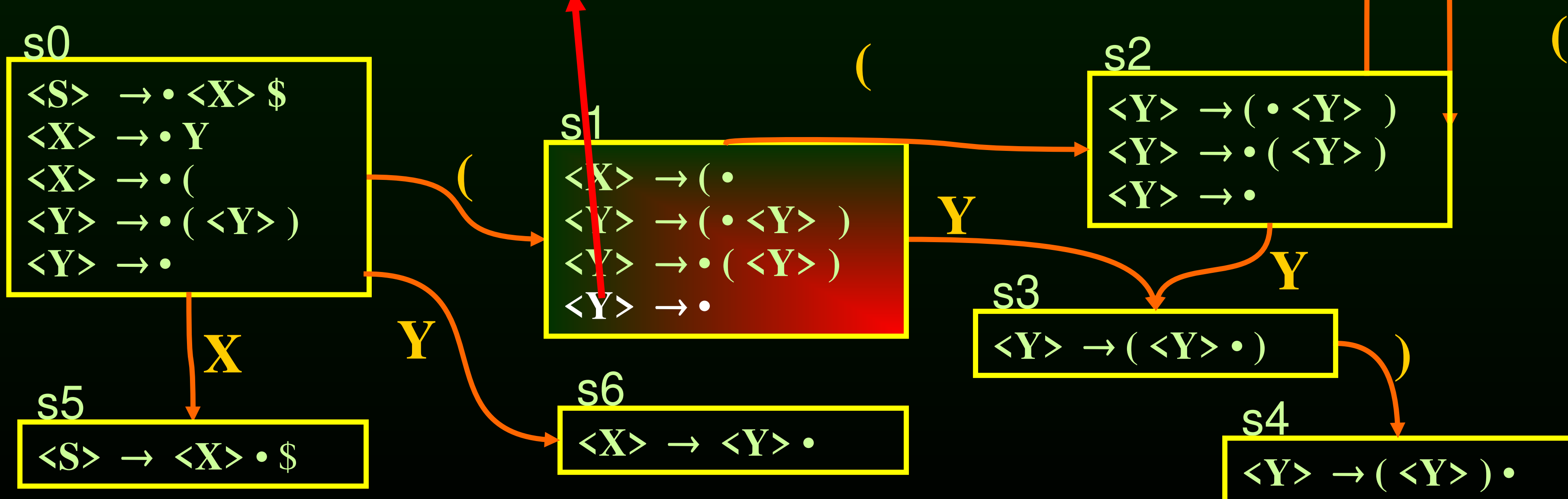


	ACTION			Goto	
State	(	)	$	X	Y
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2		reduce (3)		goto s3
s2					
s3					
s4					
s5					
s6					



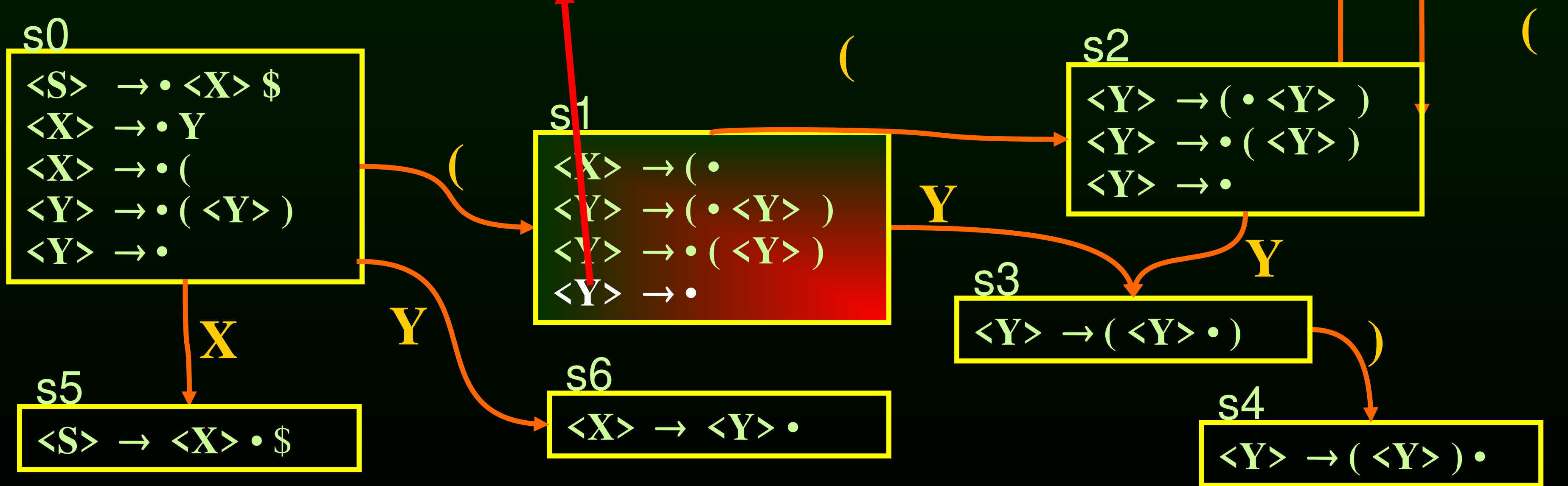
	ACTION			Goto	
State	(	)	$	X	Y
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2		reduce (3)		goto s3
s2					
s3					
s4					
s5					
s6					

$$\text{Follow}(\langle Y\rangle) = \{\ ),\ \$\ \}$$



	ACTION			Goto	
State	(	)	$	X	Y
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	reduce (3)		goto s3
s2					
s3					
s4					
s5					
s6					

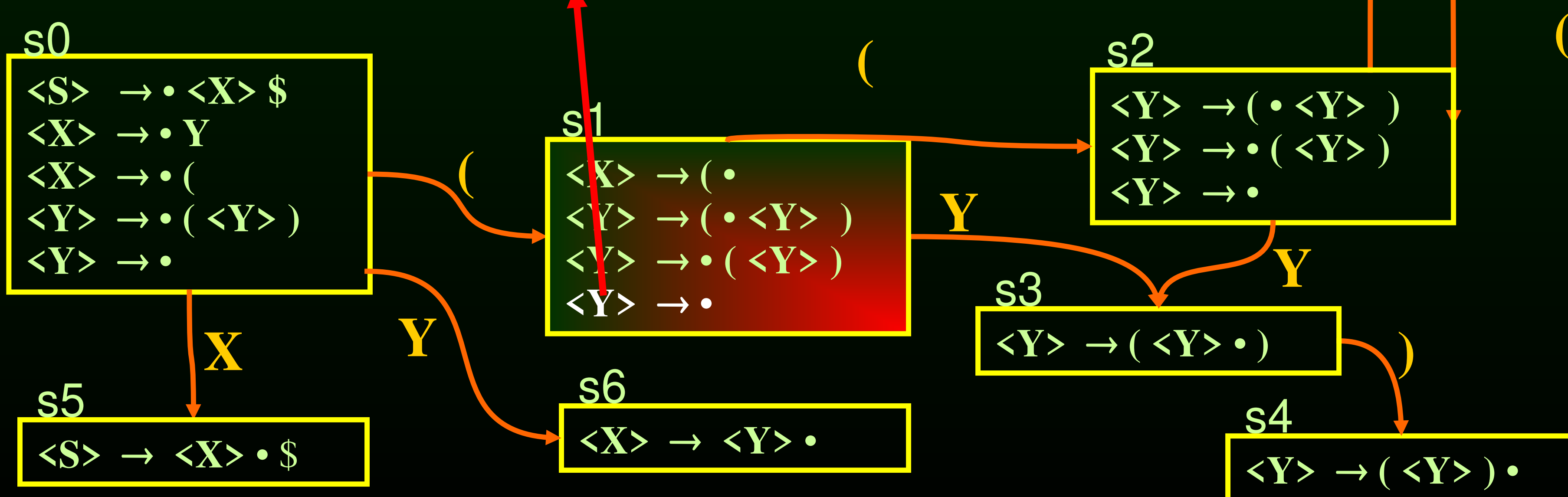
Follow(<Y>) = {), $ }



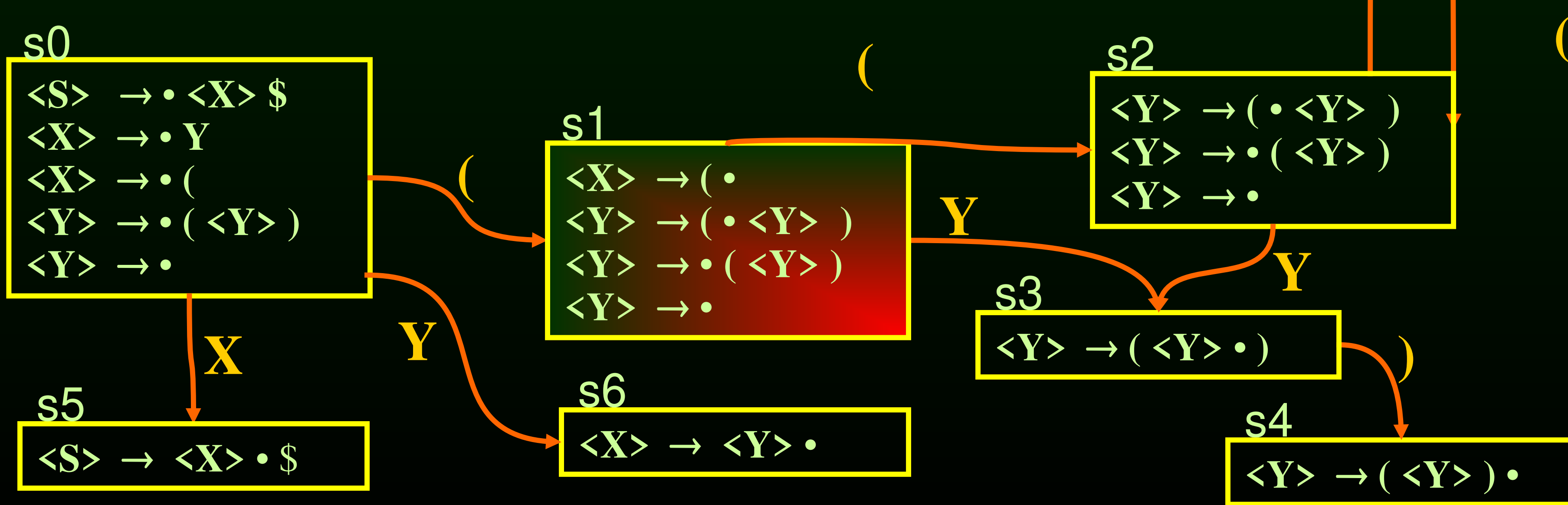
	ACTION			Goto	
State	(	)	$	X	Y
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	**reduce(3 or 5)**		goto s3
s2					
s3					
s4					
s5					
s6					

Follow(<Y>) = {), $ }

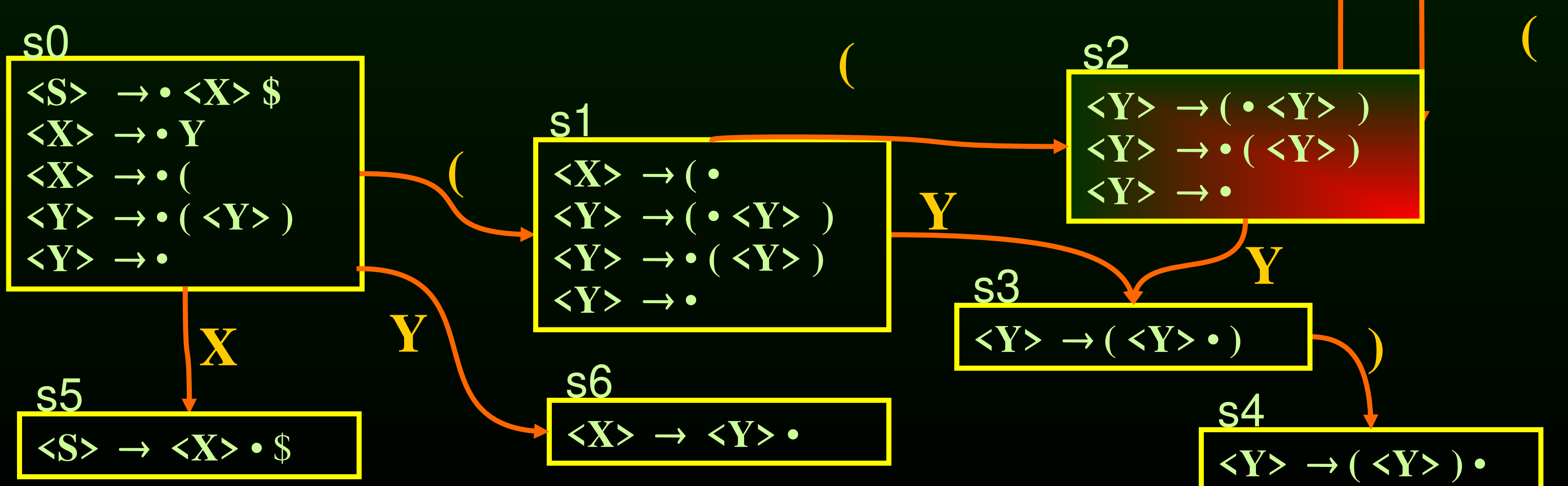
¡Conflicto reduce/reduce!



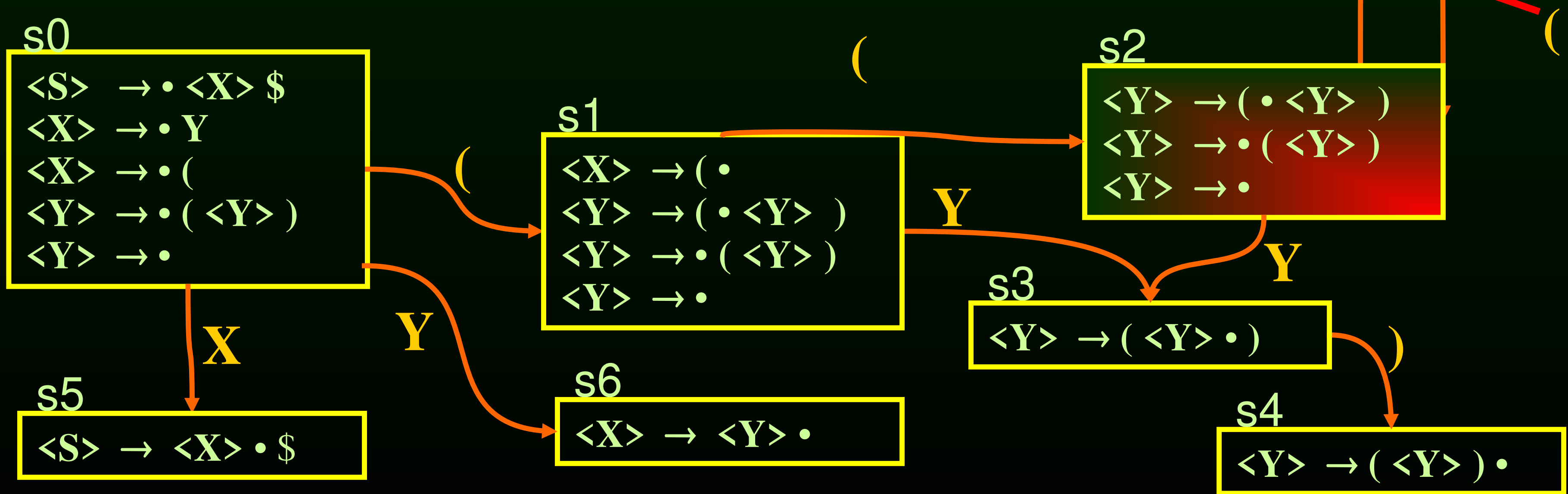
	ACTION			Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	**reduce(3 or 5)**		goto s3
s2					
s3					
s4					
s5					
s6					



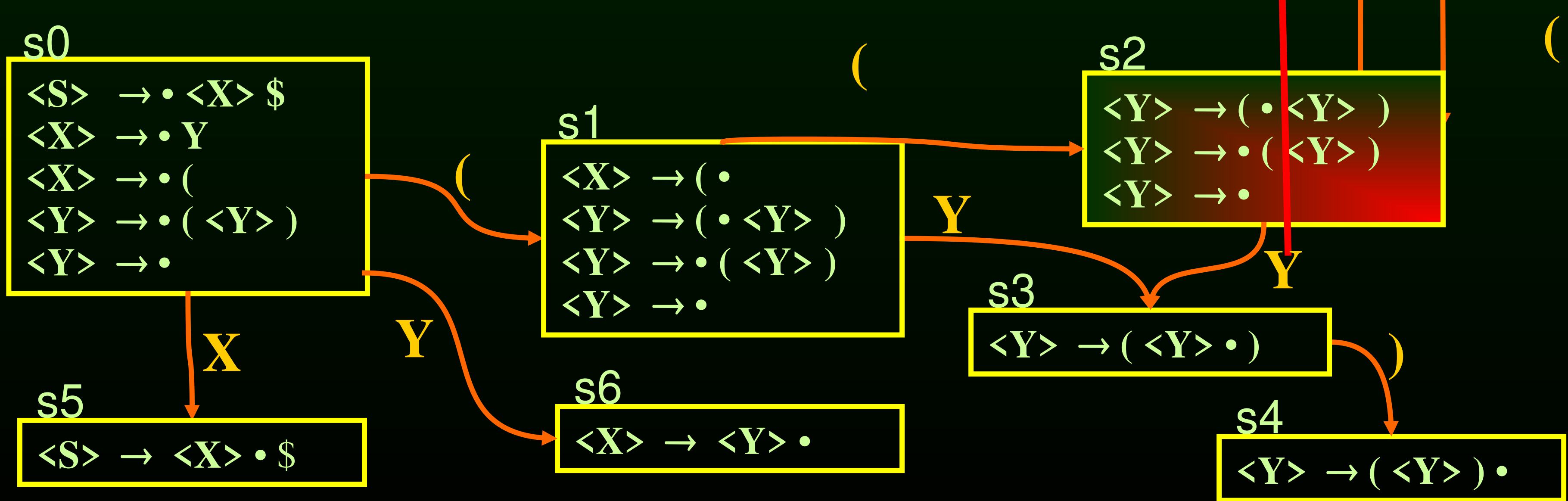
	ACTION			Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	**reduce(3 or 5)**		goto s3
s2					
s3					
s4					
s5					
s6					



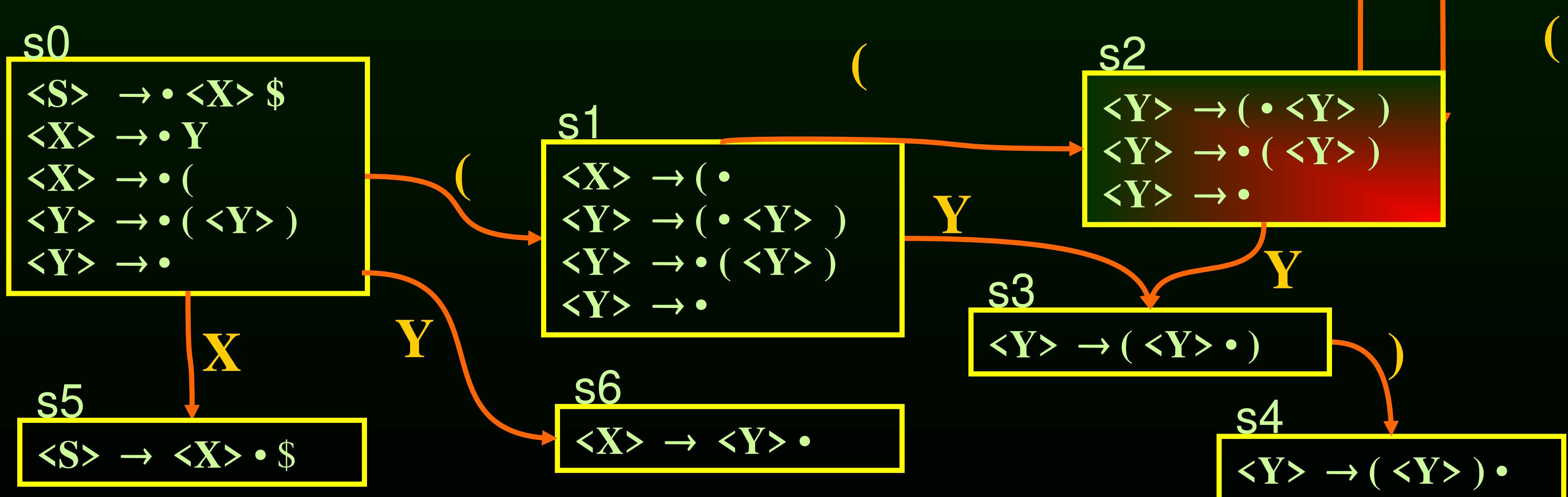
	ACTION			Goto	
State	(	)	$	X	Y
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	**reduce(3 or 5)**		goto s3
s2	shift to s2				
s3					
s4					
s5					
s6					



		ACTION		Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	**reduce(3 or 5)**		goto s3
s2	shift to s2				goto s3
s3					
s4					
s5					
s6					

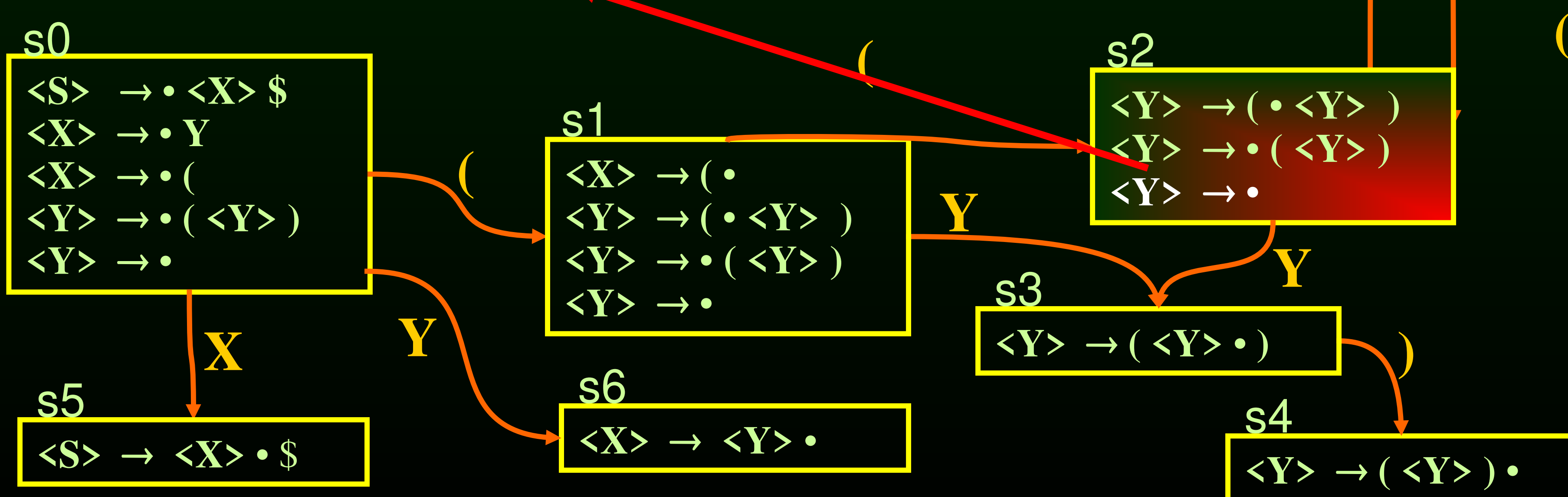


		ACTION		Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	**reduce(3 or 5)**		goto s3
s2	shift to s2				goto s3
s3					
s4					
s5					
s6					



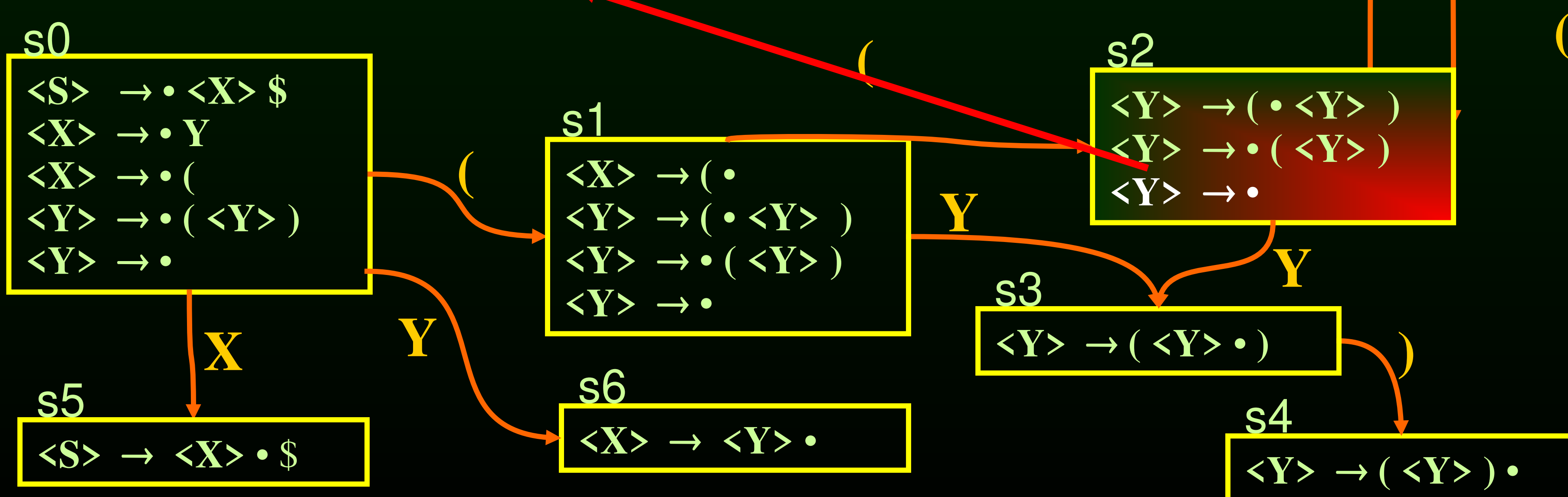
	ACTION			Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	**reduce(3 or 5)**		goto s3
s2	shift to s2				goto s3
s3					
s4					
s5					
s6					

$$\text{follow}(\langle Y \rangle) = \{\ ),\ \$\ \}$$



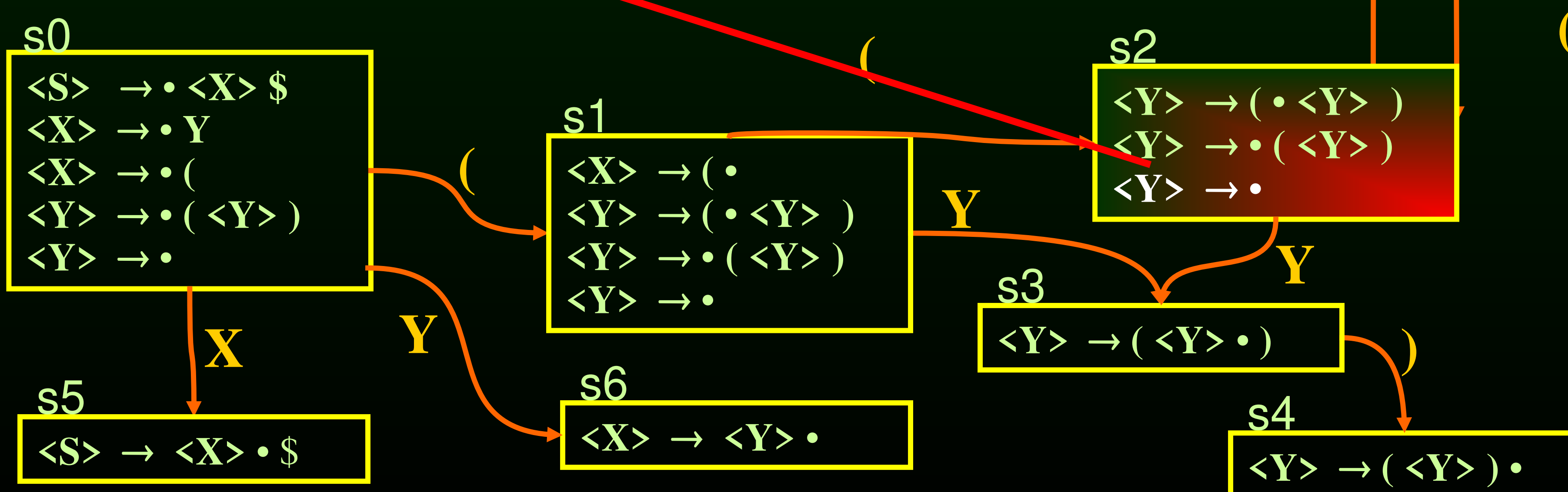
		ACTION		Goto	
State	(	)	$	X	Y
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	**reduce(3 or 5)**		goto s3
s2	shift to s2	reduce (5)			goto s3
s3					
s4					
s5					
s6					

follow(<Y>) = {), $ }

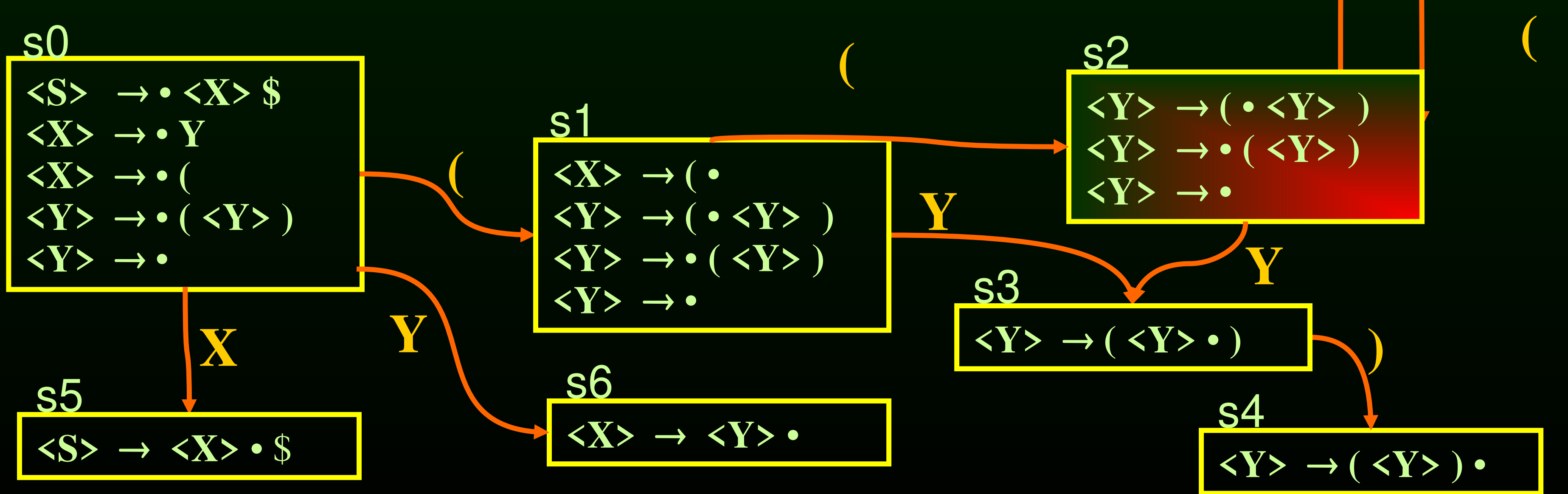


	ACTION			Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	**reduce(3 or 5)**		goto s3
s2	shift to s2	reduce (5)	reduce (5)		goto s3
s3					
s4					
s5					
s6					

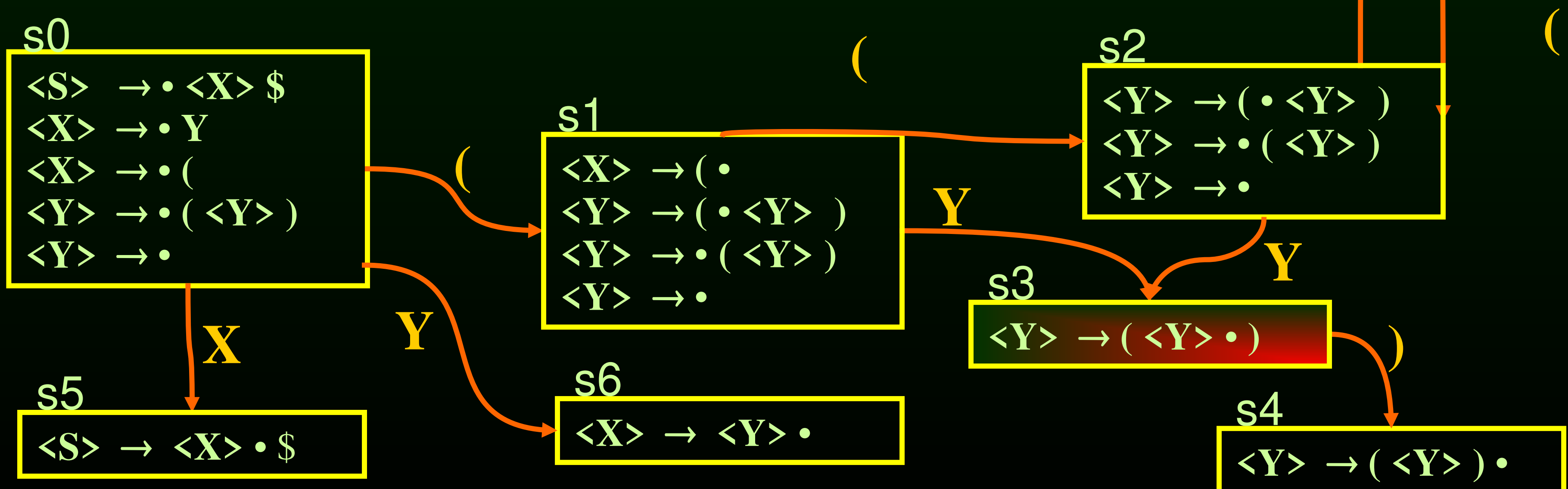
follow(<Y>) = {), $ }



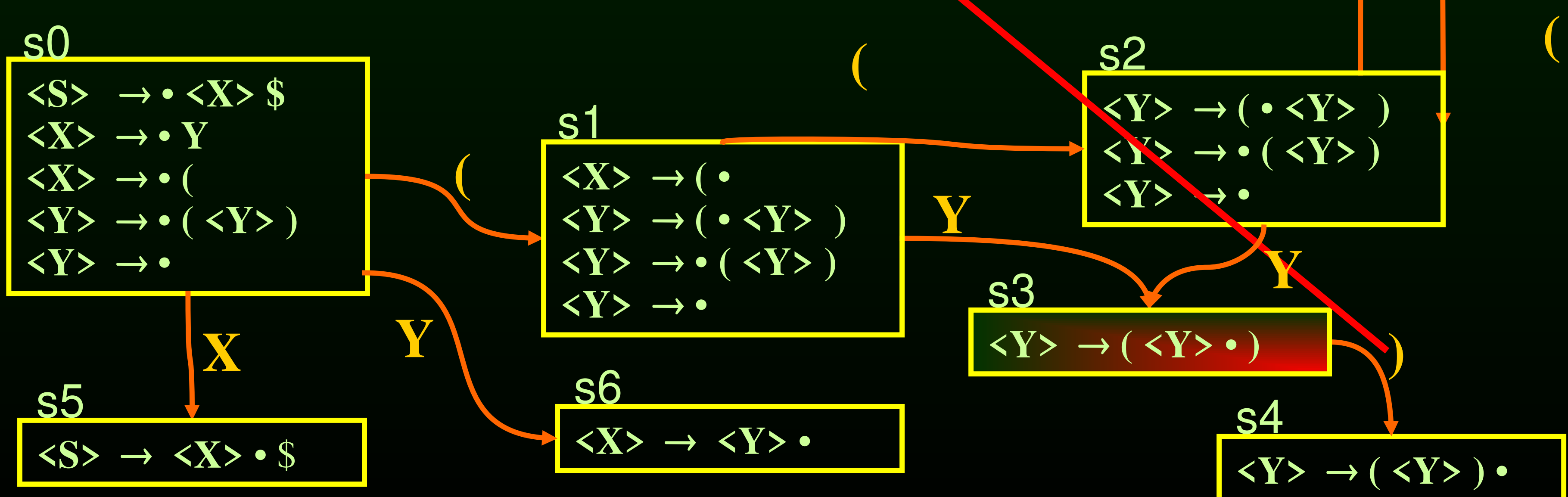
	ACTION			Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	**reduce(3 or 5)**		goto s3
s2	shift to s2	reduce (5)	reduce (5)		goto s3
s3					
s4					
s5					
s6					



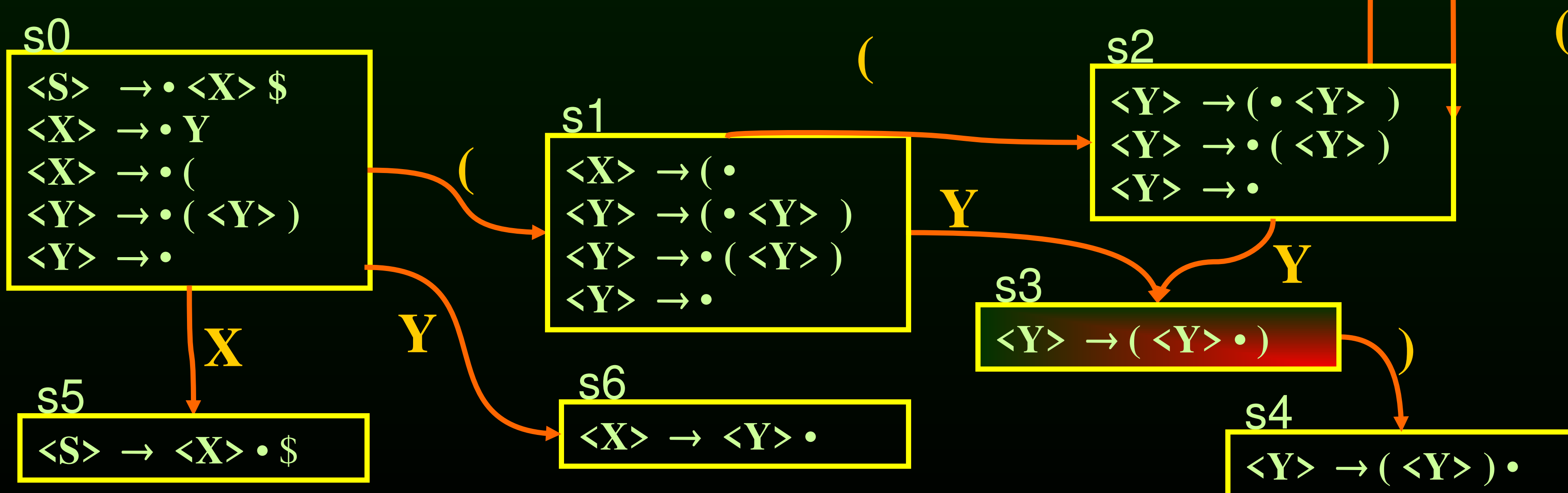
	ACTION			Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	**reduce(3 or 5)**		goto s3
s2	shift to s2	reduce (5)	reduce (5)		goto s3
s3					
s4					
s5					
s6					



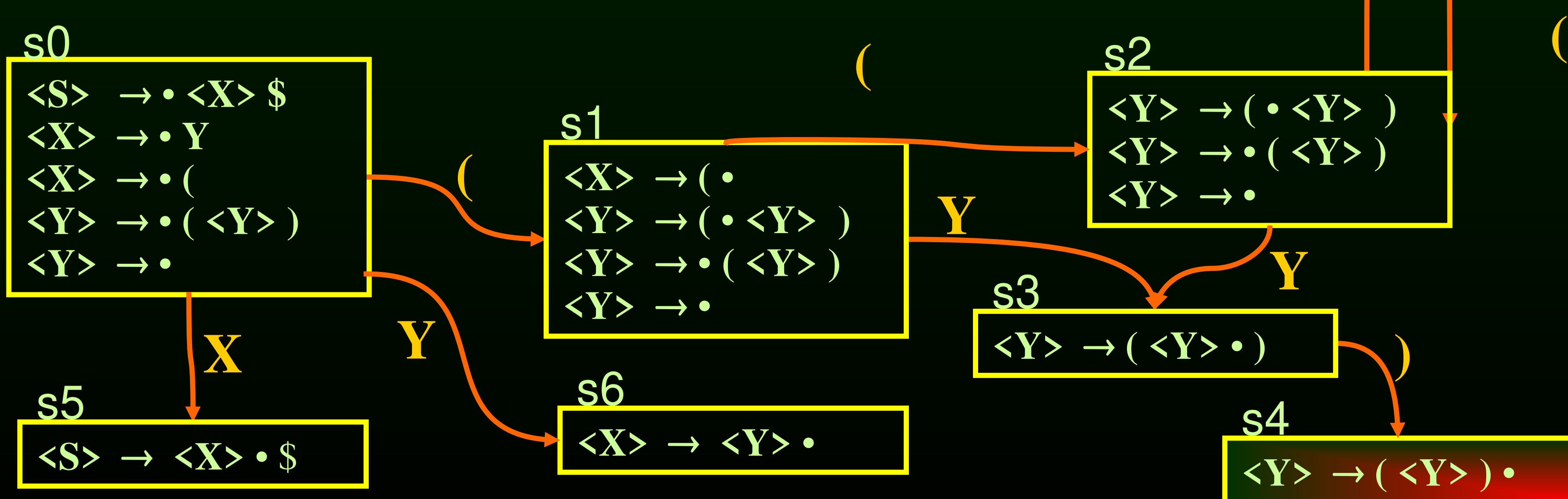
		ACTION		Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	**reduce(3 or 5)**		goto s3
s2	shift to s2	reduce (5)	reduce (5)		goto s3
s3		shift to s4			
s4					
s5					
s6					



		ACTION		Goto	
State	(	)	$	X	Y
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	**reduce(3 or 5)**		goto s3
s2	shift to s2	reduce (5)	reduce (5)		goto s3
s3	error	shift to s4	error		
s4					
s5					
s6					

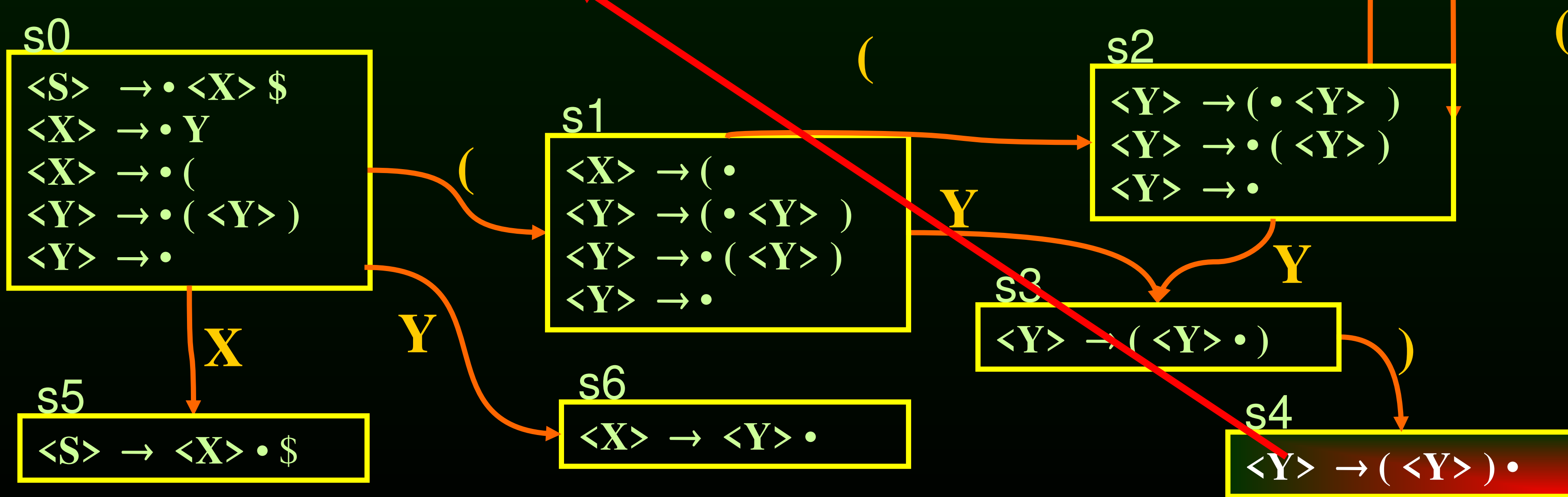


		ACTION		Goto	
State	(	)	$	X	Y
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	**reduce(3 or 5)**		goto s3
s2	shift to s2	reduce (5)	reduce (5)		goto s3
s3	error	shift to s4	error		
s4					
s5					
s6					



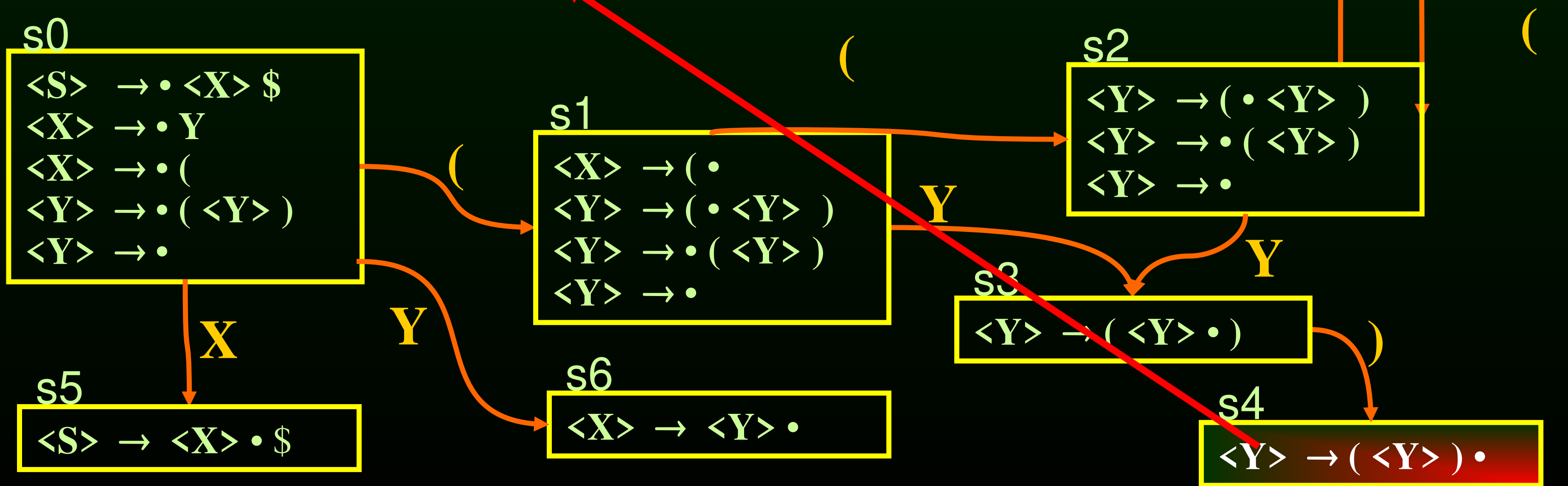
		ACTION		Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	**reduce(3 or 5)**		goto s3
s2	shift to s2	reduce (5)	reduce (5)		goto s3
s3	error	shift to s4	error		
s4					
s5					
s6					

follow(<Y>) = {), $ }



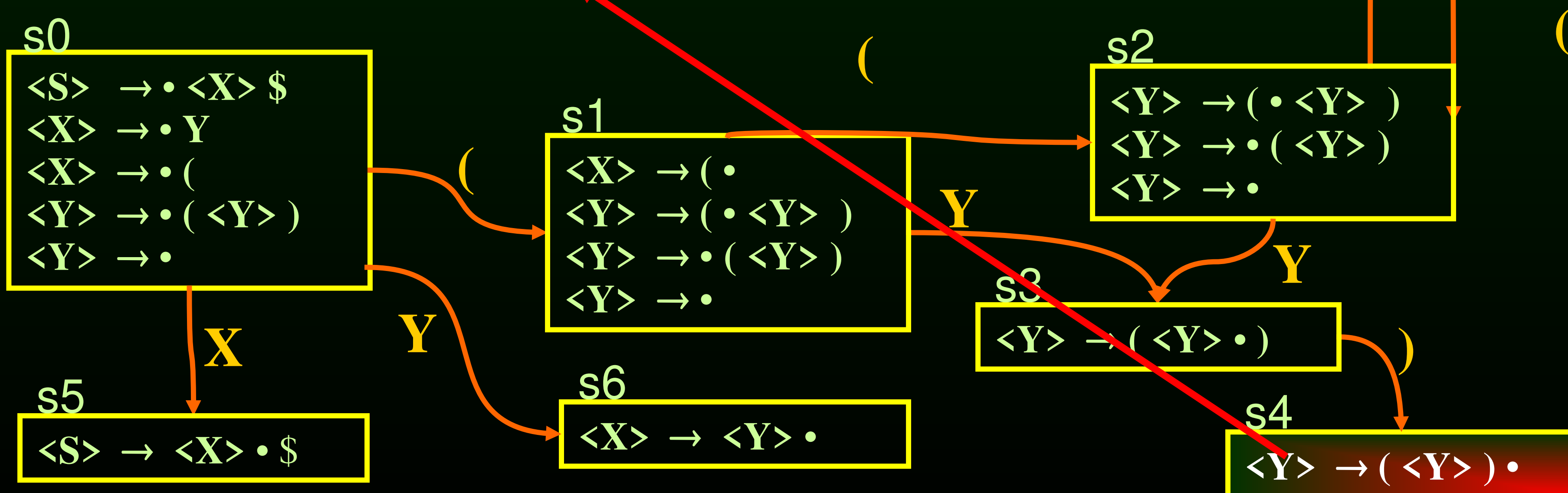
		ACTION		Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	**reduce(3 or 5)**		goto s3
s2	shift to s2	reduce (5)	reduce (5)		goto s3
s3	error	shift to s4	error		
s4		reduce (4)			
s5					
s6					

follow(<Y>) = {), $ }

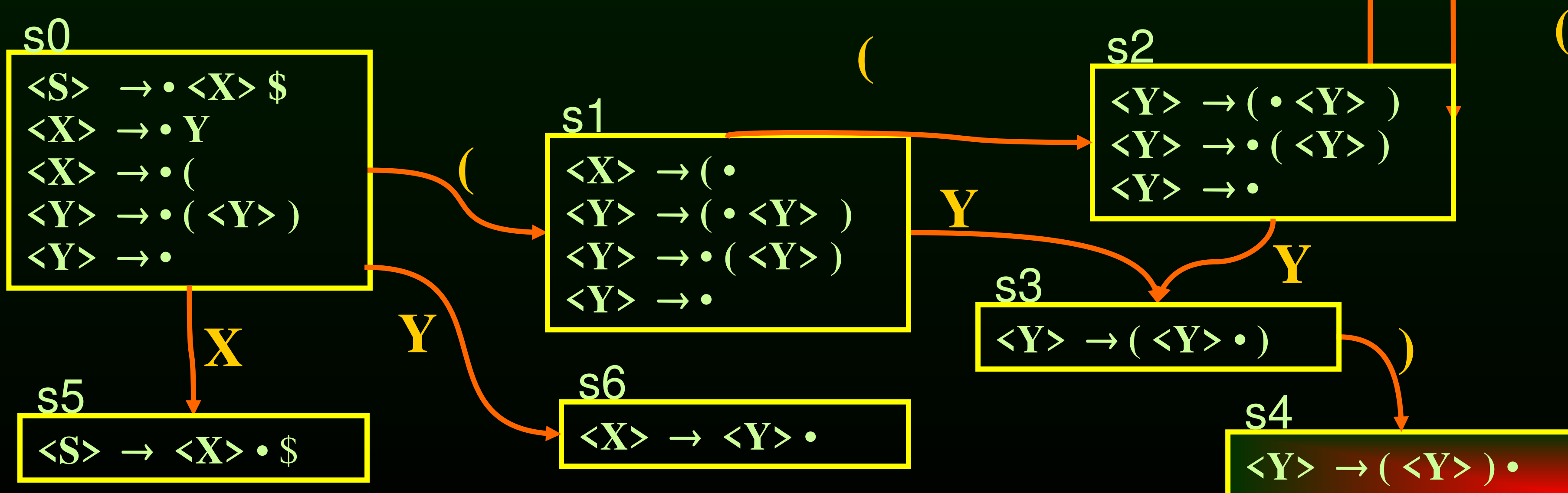


		ACTION		Goto	
State	(	)	$	X	Y
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	**reduce(3 or 5)**		goto s3
s2	shift to s2	reduce (5)	reduce (5)		goto s3
s3	error	shift to s4	error		
s4		reduce (4)	reduce (4)		
s5					
s6					

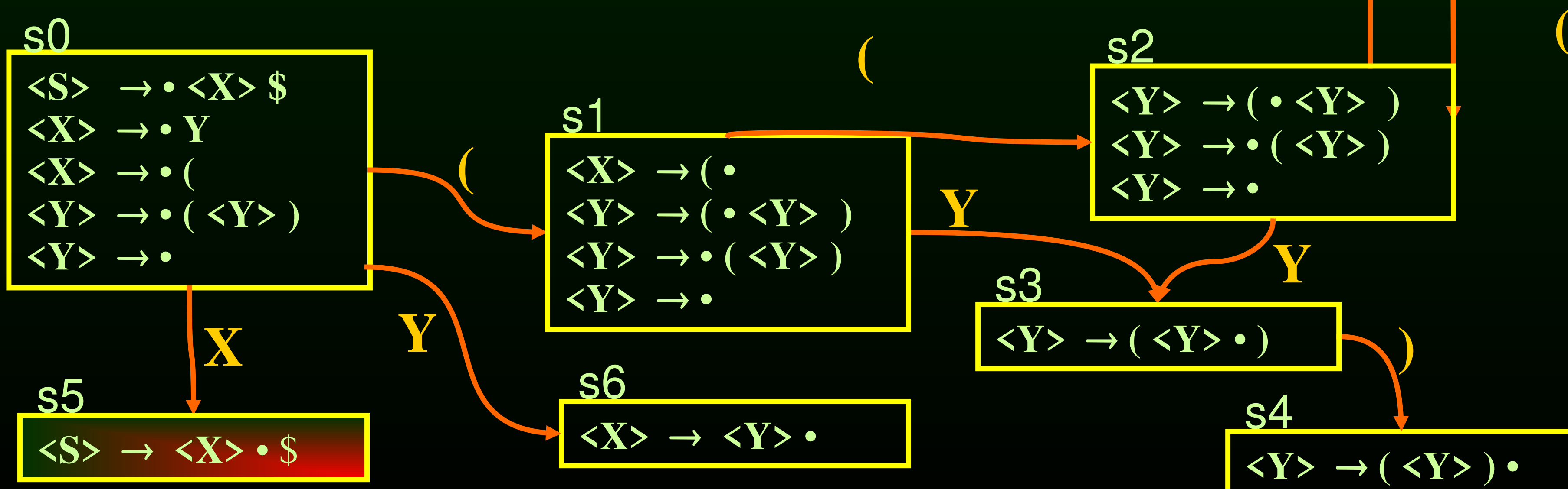
follow(<Y>) = {), $ }



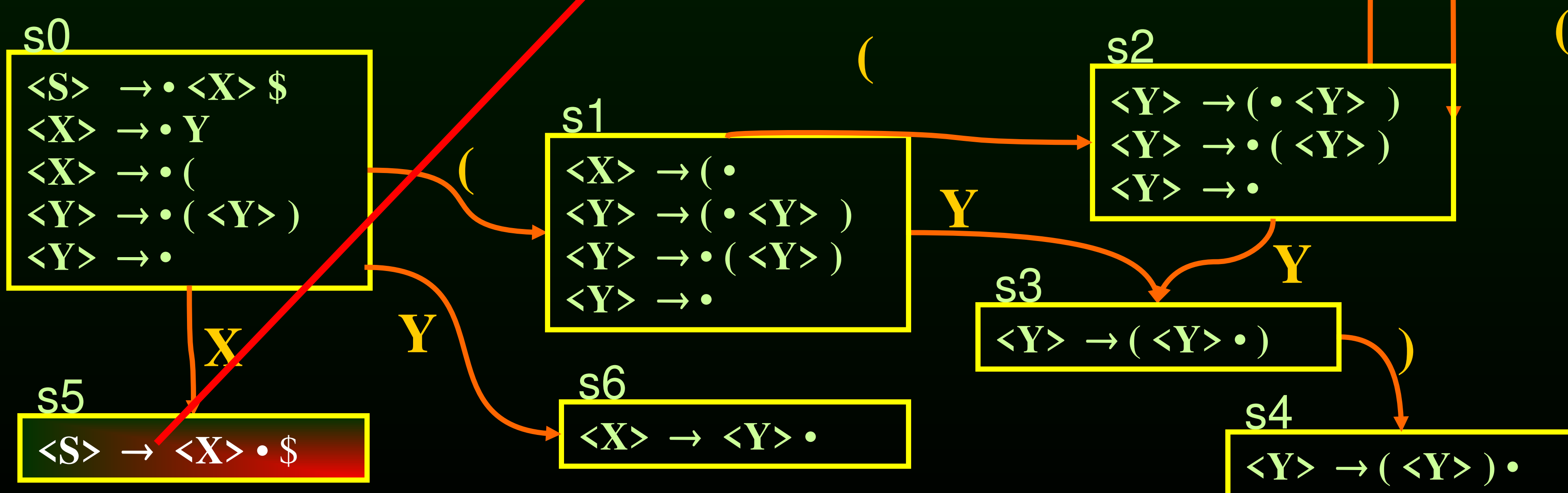
	ACTION			Goto	
State	(	)	$	X	Y
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	**reduce(3 or 5)**		goto s3
s2	shift to s2	reduce (5)	reduce (5)		goto s3
s3	error	shift to s4	error		
s4	error	reduce (4)	reduce (4)		
s5					
s6					



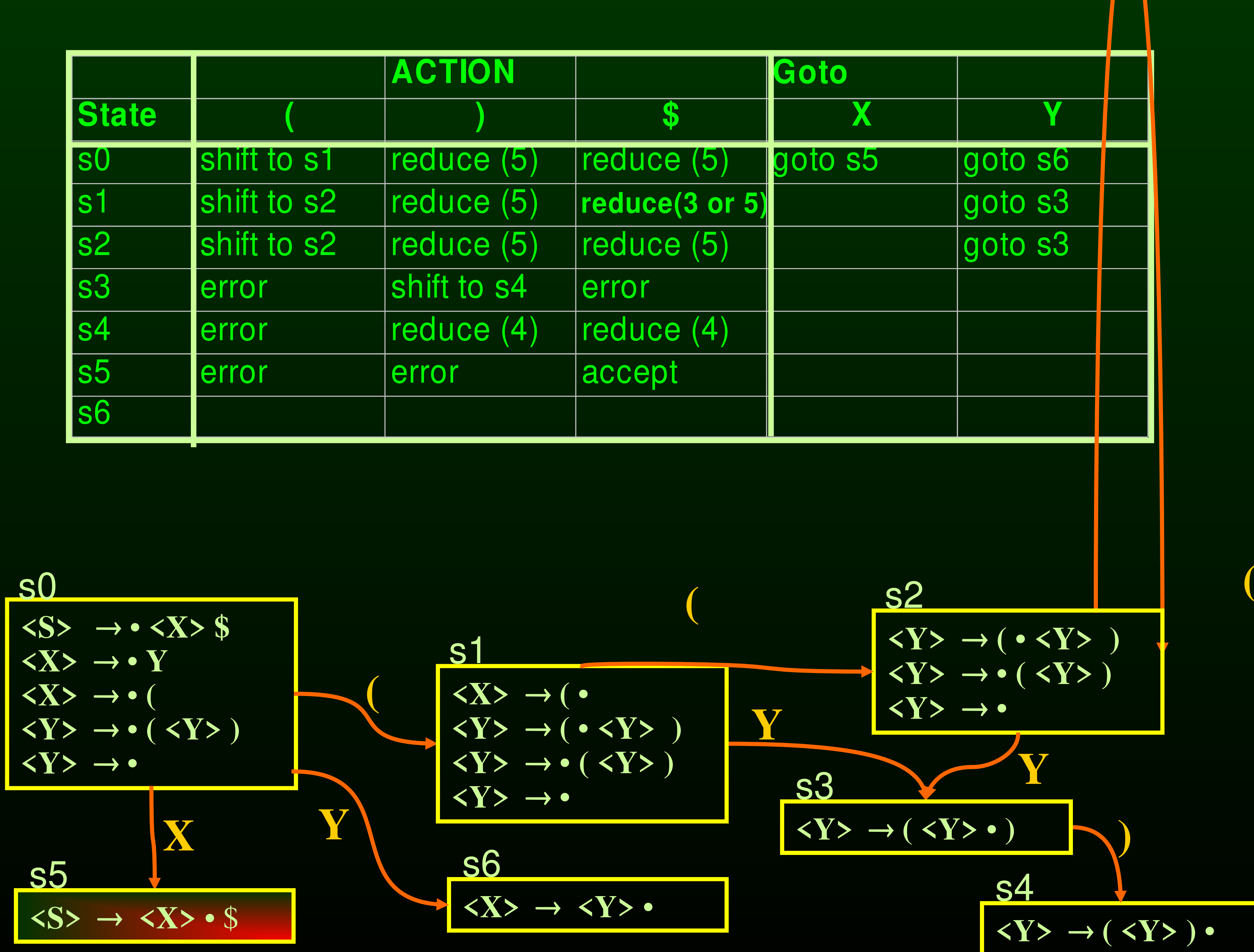
State	ACTION (	)	$	Goto X	Y
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	**reduce(3 or 5)**		goto s3
s2	shift to s2	reduce (5)	reduce (5)		goto s3
s3	error	shift to s4	error		
s4	error	reduce (4)	reduce (4)		
s5					
s6					



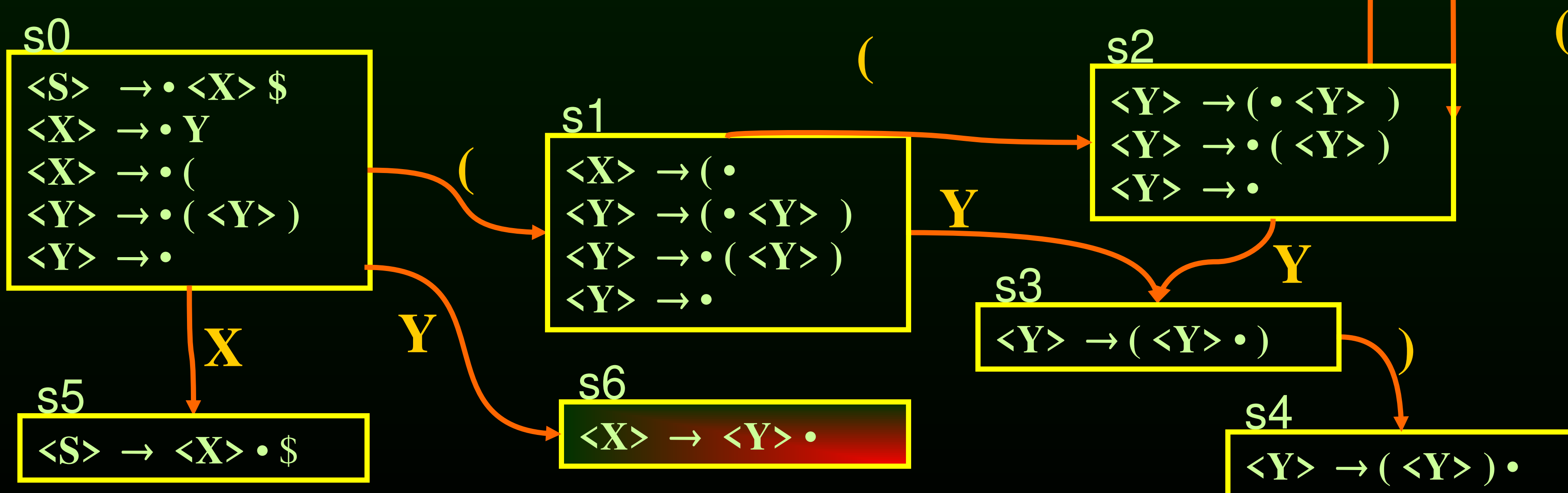
		ACTION		Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	**reduce(3 or 5)**		goto s3
s2	shift to s2	reduce (5)	reduce (5)		goto s3
s3	error	shift to s4	error		
s4	error	reduce (4)	reduce (4)		
s5			accept		
s6					



	ACTION			Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	**reduce(3 or 5)**		goto s3
s2	shift to s2	reduce (5)	reduce (5)		goto s3
s3	error	shift to s4	error		
s4	error	reduce (4)	reduce (4)		
s5	error	error	accept		
s6					

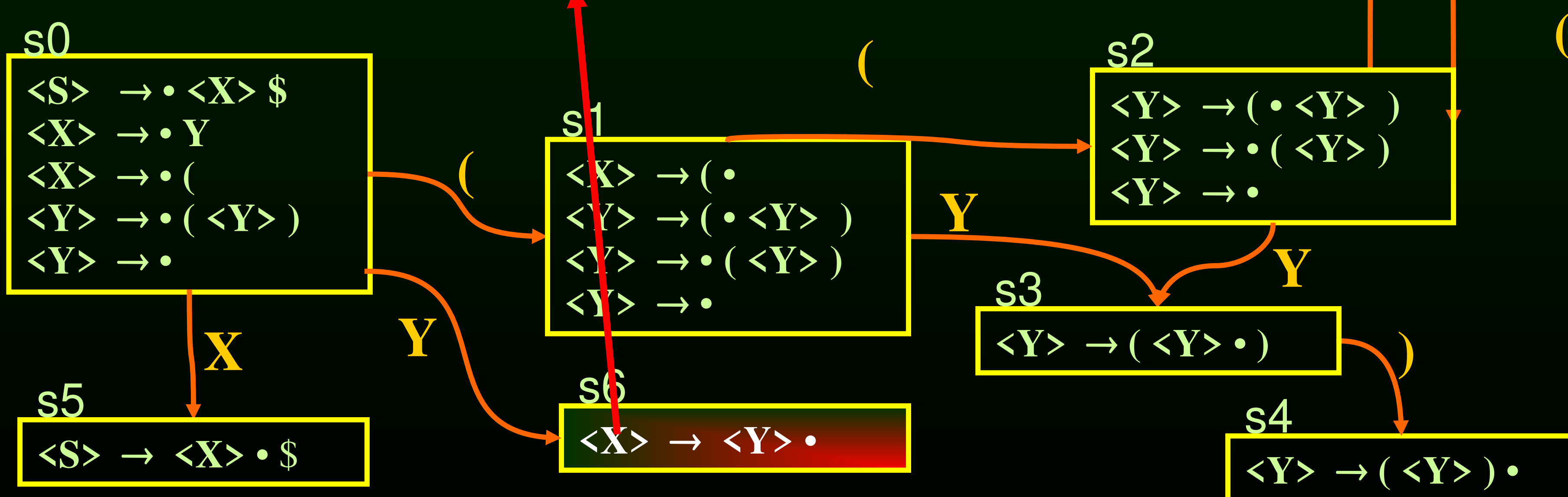


	ACTION			Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	**reduce(3 or 5)**		goto s3
s2	shift to s2	reduce (5)	reduce (5)		goto s3
s3	error	shift to s4	error		
s4	error	reduce (4)	reduce (4)		
s5	error	error	accept		
s6					



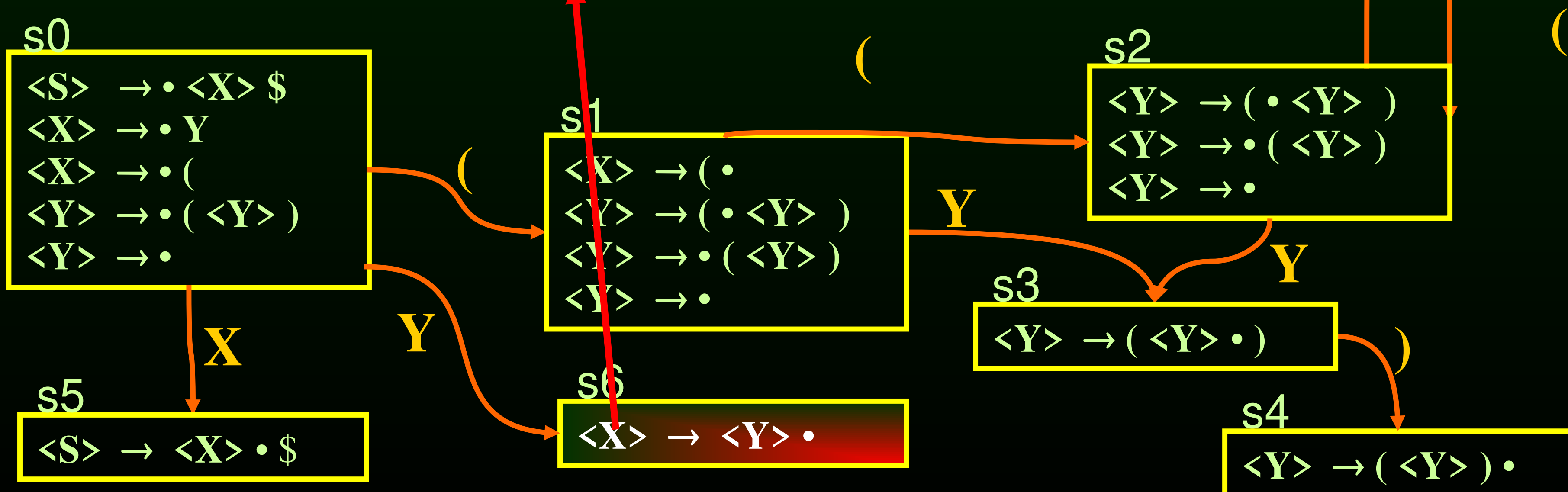
	ACTION			Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	**reduce(3 or 5)**		goto s3
s2	shift to s2	reduce (5)	reduce (5)		goto s3
s3	error	shift to s4	error		
s4	error	reduce (4)	reduce (4)		
s5	error	error	accept		
s6					

follow(<X>) = { $ }

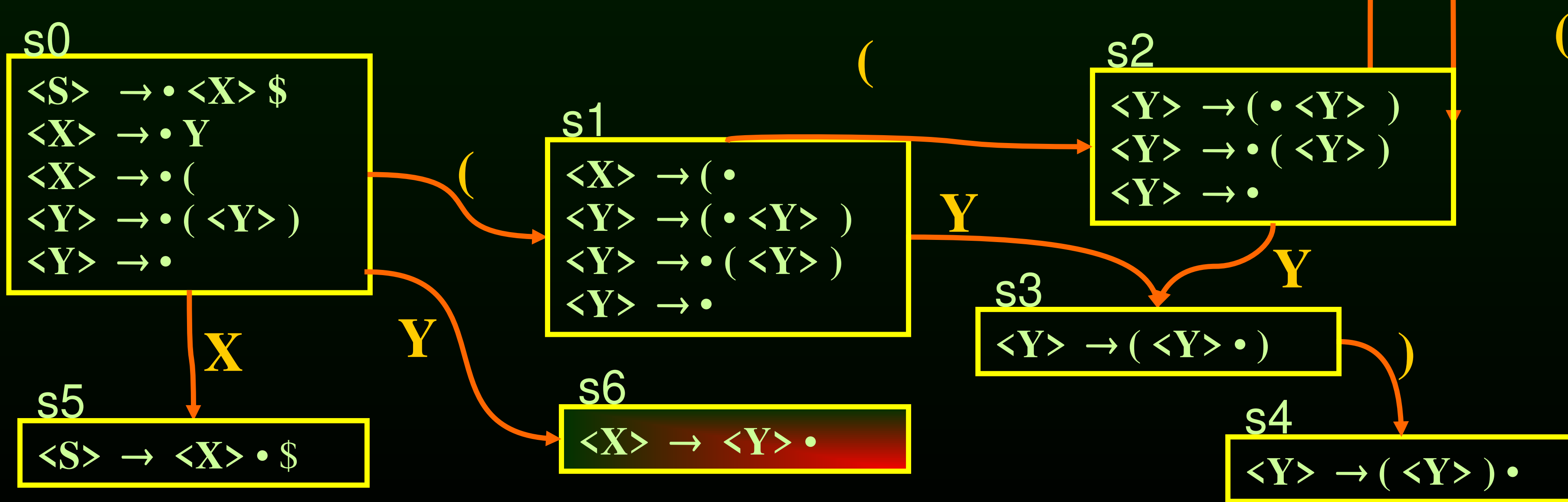


		ACTION		Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	**reduce(3 or 5)**		goto s3
s2	shift to s2	reduce (5)	reduce (5)		goto s3
s3	error	shift to s4	error		
s4	error	reduce (4)	reduce (4)		
s5	error	error	accept		
s6			reduce (2)		

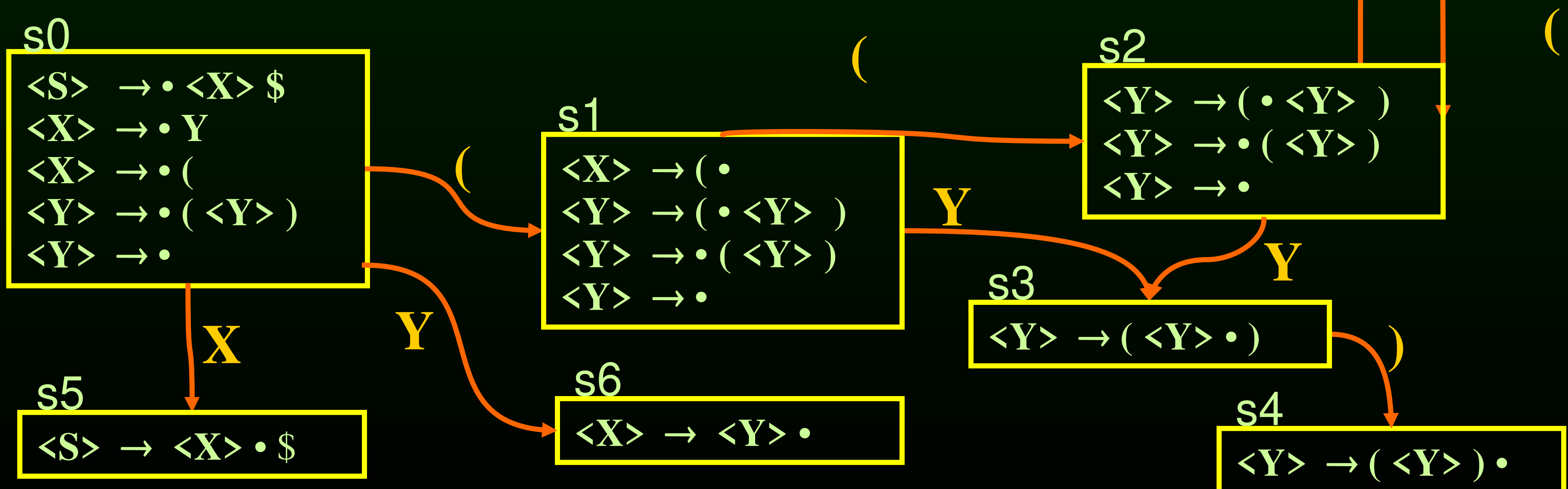
follow(<X>) = { $ }



	ACTION			Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	**reduce(3 or 5)**		goto s3
s2	shift to s2	reduce (5)	reduce (5)		goto s3
s3	error	shift to s4	error		
s4	error	reduce (4)	reduce (4)		
s5	error	error	accept		
s6	error	error	reduce (2)		



	ACTION			Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	**reduce(3 or 5)**		goto s3
s2	shift to s2	reduce (5)	reduce (5)		goto s3
s3	error	shift to s4	error		
s4	error	reduce (4)	reduce (4)		
s5	error	error	accept		
s6	error	error	reduce (2)		



		ACTION		Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	**reduce(3 or 5)**		goto s3
s2	shift to s2	reduce (5)	reduce (5)		goto s3
s3	error	shift to s4	error		
s4	error	reduce (4)	reduce (4)		
s5	error	error	accept		
s6	error	error	reduce (2)		

¿Podemos eliminar los conflictos reduce/reduce?

Resumen

- Resumen de la construcción de un parser LR(0)
- Limitaciones de los lenguajes LR(0)
- Construcción de un parser engine SLR(1)
- Limitaciones de los lenguajes SLR(1)
- Construcción de un parser engine LR(1)
- Construcción de un parser engine LALR(1)

Items LR(1)

- Los ítems mantienen información acerca de
  - producciones
  - posición en el lado derecho (el punto)
  - símbolo de look ahead
- Un ítem LR(1) es de la forma $[\mathbf{A \rightarrow \alpha \bullet \beta \quad a}]$
  - $\mathbf{A \rightarrow \alpha\, \beta}$ es una producción
  - El punto en $\mathbf{A \rightarrow \alpha \bullet \beta}$ denota la posición
  - **a** es un terminal o el símbolo de EOF ($)
- Para el ítem $[\mathbf{A \rightarrow \alpha \bullet \quad a}]$
  - **a** es el siguiente símbolo después de **A** en el string, es decir, existe una derivación $\mathbf{S \Rightarrow \gamma\, A\, a\, \lambda}$

- La gramática

  <S> → <X> $

  <X> → (<X>)

  <X> → ε

- Símbolos terminales
  - ‘(‘ ‘)’
- Fin del input
  - ‘$’

Items LR(1)

[<S> → • <X> $)]

[<S> → • <X> $ ( ]

[<S> → • <X> $ $]

[<S> → <X> • $)]

[<S> → <X> • $ ( ]

[<S> → <X> • $ $]

[<X> → • (<X>))]

[<X> → • (<X>) ( ]

[<X> → • (<X>) $]

[<X> → (• <X>))]

[<X> → (• <X>) ( ]

[<X> → (• <X>) $]

[<X> → (<X> •))]

[<X> → (<X> •) ( ]

[<X> → (<X> •) $]

[<X> → (<X>) •)]

[<X> → (<X>) • ( ]

[<X> → (<X>) • $]

[<X> → •)]

[<X> → • ( ]

[<X> → • $]

Calculando los ítems LR(1) y los conjuntos first() y follow()

- Para cualquier string α, **first(α)** es el conjunto de terminales que comienzan un string derivado de α

$$\text{first}(\alpha) = \left\{ c \mid \exists \text{derivacion} \quad \alpha \Rightarrow c\beta \right\}$$

- Para cada no-terminal A, **follow(A)** es el conunto de terminales que pueden ocurrir después de A

$$\text{follow}(A) = \left\{ b \mid \exists \text{derivacion} \quad S \Rightarrow \alpha A b \beta \right\}$$

Algoritmo para first()

- Si a es un terminal
  entonces first(a) = { a }
- Si $\mathbf{A} \rightarrow \boldsymbol{\varepsilon}$ es una producción
  o $\mathbf{A} \rightarrow \mathbf{X_1\, X_2 \ldots\, X_k}$ es una producción
  y $\varepsilon \in$ first(X_1), …, first(X_k)
  entonces first(A) = first(A) $\cup$ { ε }
- Si $\mathbf{A} \rightarrow \mathbf{X_1\, X_2 \ldots\, X_i\, X_{i+1} \ldots\, X_k}$ es una producción
  y $\varepsilon \in$ first(X_1), …, first(X_i)
  y el terminal a $\in$ first(X_{i+1})
  entonces first(A) = first(A) $\cup$ { a }

Ejemplo de first()

- Gramática

<S> → <X> $
<X> → <Y>
<X> → (
<Y> → (<Y>)
<Y> → ε

Ejemplo de first()

- Gramática

  <S> → <X> $
  <X> → <Y>
  <X> → (
  <Y> → (<Y>)
  <Y> → ε

  first( )) = { }
  first(( ) = { }
  first(S) = { }
  first(X) = { }
  first(Y) = { }

Ejemplo de first()

- Gramática

  <S> → <X> $

  <X> → <Y>

  <X> → (

  <Y> → (<Y>)

  <Y> → ε

  first( )) = { }

  first(( ) = { }

  first(S) = { }

  first(X) = { }

  first(Y) = { }

- Si a es un terminal
  entonces first(a) = { a }

- Si **A → ε** es una producción
  o **A → X_1 X_2... X_k** es una producción
  y ε ∈ first(X_1), …, first(X_k)
  entonces first(A) = first(A) ∪ { ε }

- Si **A → X_1 X_2... X_i X_{i+1} ... X_k** es una producción
  y ε ∈ first(X_1), …, first(X_i)
  y el terminal a ∈ first(X_{i+1})
  entonces first(A) = first(A) ∪ { a }

Ejemplo de first()

- Grammar

  <S> → <X> $
  <X> → <Y>
  <X> → (
  <Y> → (<Y>)
  <Y> → ε

  first( )) = {) }
  first(( ) = { }
  first(S) = { }
  first(X) = { }
  first(Y) = { }

- Si a es un terminal
  entonces first(a) = { a }

- Si **A →** **ε** es una producción
  o **A →** $\mathbf{X_1\, X_2 \ldots X_k}$ es una producción
  y ε ∈ first(X_1), …, first(X_k)
  entonces first(A) = first(A) ∪ { ε }

- Si **A →** $\mathbf{X_1\, X_2 \ldots X_i\; X_{i+1} \ldots X_k}$ es una producción
  y ε ∈ first(X_1), …, first(X_i)
  y el terminal a ∈ first(X_{i+1})
  entonces first(A) = first(A) ∪ { a }

Ejemplo de first()

- Grammar

  <S> → <X> $

  <X> → <Y>

  <X> → (

  <Y> → (<Y>)

  <Y> → ε

first( )) = {) }

first(( ) = { }

first(S) = { }

first(X) = { }

first(Y) = { }

- Si a es un terminal
  entonces first(a) = { a }

- Si **A → ε** es una producción
  o **A →** $\mathbf{X_1\, X_2 \ldots X_k}$ es una producción
  y $\varepsilon \in \text{first}(X_1), \ldots, \text{first}(X_k)$
  entonces $\text{first}(A) = \text{first}(A) \cup \{ \varepsilon \}$

- Si **A →** $\mathbf{X_1\, X_2 \ldots X_i\, X_{i+1} \ldots X_k}$ es una producción
  y $\varepsilon \in \text{first}(X_1), \ldots, \text{first}(X_i)$
  y el terminal $a \in \text{first}(X_{i+1})$
  entonces $\text{first}(A) = \text{first}(A) \cup \{ a \}$

Ejemplo de first()

- Grammar

  $\langle S \rangle \rightarrow \langle X \rangle \ \$$

  $\langle X \rangle \rightarrow \langle Y \rangle$

  $\langle X \rangle \rightarrow ($

  $\langle Y \rangle \rightarrow (\ \langle Y \rangle \ )$

  $\langle Y \rangle \rightarrow \varepsilon$

  first( )) = {) }

  first(( ) = { (}

  first(S) = { }

  first(X) = { }

  first(Y) = { }

- Si a es un terminal
  entonces first(a) = { a }

- Si $\mathbf{A} \rightarrow \boldsymbol{\varepsilon}$ es una producción
  o $\mathbf{A} \rightarrow \mathbf{X_1\, X_2 \ldots\, X_k}$ es una producción
  y $\varepsilon \in$ first(X_1), …, first(X_k)
  entonces first(A) = first(A) $\cup$ { ε }

- Si $\mathbf{A} \rightarrow \mathbf{X_1\, X_2 \ldots\, X_i\, X_{i+1} \ldots\, X_k}$ es una producción
  y $\varepsilon \in$ first(X_1), …, first(X_i)
  y el terminal a $\in$ first(X_{i+1})
  entonces first(A) = first(A) $\cup$ { a }

Ejemplo de first()

- Grammar
  $\langle S \rangle \rightarrow \langle X \rangle\ \$$
  $\langle X \rangle \rightarrow \langle Y \rangle$
  $\langle X \rangle \rightarrow ($
  $\langle Y \rangle \rightarrow (\langle Y \rangle)$
  $\langle Y \rangle \rightarrow \varepsilon$

  first( )) = {) }
  first(( ) = { (}
  first(S) = { }
  first(X) = { }
  first(Y) = { }

- Si a es un terminal
  entonces first(a) = { a }
- Si $\mathbf{A} \rightarrow \boldsymbol{\varepsilon}$ es una producción
  o $\mathbf{A} \rightarrow \mathbf{X_1\, X_2 \ldots\, X_k}$ es una producción
  y $\varepsilon \in$ first(X_1), …, first(X_k)
  entonces first(A) = first(A) $\cup$ { ε }
- Si $\mathbf{A} \rightarrow \mathbf{X_1\, X_2 \ldots\, X_i\, X_{i+1} \ldots\, X_k}$ es una producción
  y $\varepsilon \in$ first(X_1), …, first(X_i)
  y el terminal a $\in$ first(X_{i+1})
  entonces first(A) = first(A) $\cup$ { a }

Ejemplo de first()

- Grammar

  <S> → <X> $

  <X> → <Y>

  <X> → (

  <Y> → (<Y>)

  <Y> → ε

first( )) = {) }

first(( ) = { (}

first(S) = { }

first(X) = { }

first(Y) = { ε }

- Si a es un terminal
  entonces first(a) = { a }

- Si **A →** **ε** es una producción
  o **A →** $\mathbf{X_1 X_2 ... X_k}$ es una producción
  y ε ∈ first(X_1), …, first(X_k)
  entonces first(A) = first(A) ∪ { ε }

- Si **A →** $\mathbf{X_1 X_2 ... X_i\ X_{i+1} ... X_k}$ es una producción
  y ε ∈ first(X_1), …, first(X_i)
  y el terminal a ∈ first(X_{i+1})
  entonces first(A) = first(A) ∪ { a }

Ejemplo de first()

- Grammar

$$\langle S \rangle \rightarrow \langle X \rangle \ \$$$
$$\langle X \rangle \rightarrow \langle Y \rangle$$
$$\langle X \rangle \rightarrow (\;$$
$$\langle Y \rangle \rightarrow (\ \langle Y \rangle\ )$$
$$\langle Y \rangle \rightarrow \varepsilon$$

$$\text{first}(\ )\ ) = \{\ )\ \}$$
$$\text{first}(\ (\ ) = \{\ (\ \}$$
$$\text{first}(S) = \{\ \ \}$$
$$\text{first}(X) = \{\ \ \}$$
$$\text{first}(Y) = \{\ \varepsilon\ \}$$

- Si a es un terminal
  entonces first(a) = { a }

- Si $\mathbf{A} \rightarrow \boldsymbol{\varepsilon}$ es una producción
  o $\mathbf{A} \rightarrow \mathbf{X_1\, X_2 \ldots\, X_k}$ es una producción
  y $\varepsilon \in \text{first}(X_1), \ldots, \text{first}(X_k)$
  entonces $\text{first}(A) = \text{first}(A) \cup \{\ \varepsilon\ \}$

- Si $\mathbf{A} \rightarrow \mathbf{X_1\, X_2 \ldots\, X_i\, X_{i+1} \ldots\, X_k}$ es una producción
  y $\varepsilon \in \text{first}(X_1), \ldots, \text{first}(X_i)$
  y el terminal $a \in \text{first}(X_{i+1})$
  entonces $\text{first}(A) = \text{first}(A) \cup \{\ a\ \}$

Ejemplo de first()

- Grammar

  <S> → <X> $
  <X> → <Y>
  <X> → (
  <Y> → (<Y>)
  <Y> → ε

first( )) = {) }
first(( ) = { (}
first(S) = { }
first(X) = { ε }
first(Y) = { ε }

- Si a es un terminal
  entonces first(a) = { a }

- Si **A → ε** es una producción
  o **A → X_1 X_2... X_k** es una producción
  y ε ∈ first(X_1), …, first(X_k)
  entonces first(A) = first(A) ∪ { ε }

- Si **A → X_1 X_2... X_i X_{i+1} ... X_k** es una producción
  y ε ∈ first(X_1), …, first(X_i)
  y el terminal a ∈ first(X_{i+1})
  entonces first(A) = first(A) ∪ { a }

Ejemplo de first()

- Grammar

  <S> → <X> $
  <X> → <Y>
  <X> → (
  <Y> → (<Y>)
  <Y> → ε

  first( )) = {) }
  first(( ) = { (}
  first(S) = { }
  first(X) = { ε }
  first(Y) = { ε }

- Si a es un terminal
  entonces first(a) = { a }

- Si **A** → **ε** es una producción
  o **A** → **X_1 X_2... X_k** es una producción
  y ε ∈ first(X_1), …, first(X_k)
  entonces first(A) = first(A) ∪ { ε }

- Si **A** → **X_1 X_2... X_i X_{i+1} ... X_k** es una producción
  y ε ∈ first(X_1), …, first(X_i)
  y el terminal a ∈ first(X_{i+1})
  entonces first(A) = first(A) ∪ { a }

Ejemplo de first()

- Grammar

  <S> → <X> $
  <X> → <Y>
  <X> → (
  <Y> → (<Y>)
  <Y> → ε

  first( )) = {) }
  first(( ) = { (}
  first(S) = { ε }
  first(X) = { ε }
  first(Y) = { ε }

- Si a es un terminal
  entonces first(a) = { a }

- Si **A → ε** es una producción
  o **A →** $\mathbf{X_1\, X_2 \ldots\, X_k}$ es una producción
  y ε ∈ first(X_1), …, first(X_k)
  entonces first(A) = first(A) ∪ { ε }

- Si **A →** $\mathbf{X_1\, X_2 \ldots\, X_i\; X_{i+1} \ldots\, X_k}$ es una producción
  y ε ∈ first(X_1), …, first(X_i)
  y el terminal a ∈ first(X_{i+1})
  entonces first(A) = first(A) ∪ { a }

Ejemplo de first()

- Grammar

  <S> → <X> $
  <X> → <Y>
  <X> → (
  <Y> → (<Y>)
  <Y> → ε

first( )) = {) }
first(( ) = { (}
first(S) = { ε }
first(X) = { ε }
first(Y) = { ε }

- Si a es un terminal
  entonces first(a) = { a }

- Si **A → ε** es una producción
  o **A → X_1 X_2... X_k** es una producción
  y ε ∈ first(X_1), …, first(X_k)
  entonces first(A) = first(A) ∪ { ε }

- Si **A → X_1 X_2... X_i X_{i+1} ... X_k** es una producción
  y ε ∈ first(X_1), …, first(X_i)
  y el terminal a ∈ first(X_{i+1})
  entonces first(A) = first(A) ∪ { a }

Ejemplo de first()

- Grammar

  <S> → <X> $
  <X> → <Y>
  <X> → (
  <Y> → (<Y>)
  <Y> → ε

first( )) = {) }
first(( ) = { (}
first(S) = { ε }
first(X) = { ε }
first(Y) = { ε, (}

- Si a es un terminal
  entonces first(a) = { a }

- Si **A → ε** es una producción
  o **A → X_1 X_2... X_k** es una producción
  y ε ∈ first(X_1), …, first(X_k)
  entonces first(A) = first(A) ∪ { ε }

- Si **A → X_1 X_2... X_i X_{i+1} ... X_k** es una producción
  y ε ∈ first(X_1), …, first(X_i)
  y el terminal a ∈ first(X_{i+1})
  entonces first(A) = first(A) ∪ { a }

Ejemplo de first()

- Grammar

  $\langle S \rangle \rightarrow \langle X \rangle \ \$$

  $\langle X \rangle \rightarrow \langle Y \rangle$

  $\langle X \rangle \rightarrow ($

  $\langle Y \rangle \rightarrow (\langle Y \rangle)$

  $\langle Y \rangle \rightarrow \varepsilon$

  first( )) = {) }

  first(( ) = { (}

  first(S) = { ε }

  first(X) = { ε }

  first(Y) = { ε, (}

- Si a es un terminal
  entonces first(a) = { a }

- Si **A →** **ε** es una producción
  o **A →** $\mathbf{X_1\, X_2 \ldots X_k}$ es una producción
  y $\varepsilon \in$ first(X_1), …, first(X_k)
  entonces first(A) = first(A) ∪ { ε }

- Si **A →** $\mathbf{X_1\, X_2 \ldots X_i\, X_{i+1} \ldots X_k}$ es una producción
  y $\varepsilon \in$ first(X_1), …, first(X_i)
  y el terminal $a \in$ first(X_{i+1})
  entonces first(A) = first(A) ∪ { a }

Ejemplo de first()

- Grammar

  <S> → <X> $

  <X> → <Y>

  <X> → (

  <Y> → (<Y>)

  <Y> → ε

first( )) = {) }

first(( ) = { (}

first(S) = { ε }

first(X) = { ε, (}

first(Y) = { ε, (}

- Si a es un terminal
  entonces first(a) = { a }

- Si **A → ε** es una producción
  o **A → X_1 X_2... X_k** es una producción
  y ε ∈ first(X_1), …, first(X_k)
  entonces first(A) = first(A) ∪ { ε }

- Si **A → X_1 X_2... X_i X_{i+1} ... X_k** es una producción
  y ε ∈ first(X_1), …, first(X_i)
  y el terminal a ∈ first(X_{i+1})
  entonces first(A) = first(A) ∪ { a }

Ejemplo de first()

- Grammar

  $\langle \mathbf{S} \rangle \rightarrow \langle \mathbf{X} \rangle \ \$$

  $\langle \mathbf{X} \rangle \rightarrow \langle \mathbf{Y} \rangle$

  $\langle \mathbf{X} \rangle \rightarrow \ ($

  $\langle \mathbf{Y} \rangle \rightarrow \ (\ \langle \mathbf{Y} \rangle \ )$

  $\langle \mathbf{Y} \rangle \rightarrow \varepsilon$

  first( )) = {) }

  first(( ) = { (}

  first(S) = { ε }

  first(X) = { ε, (}

  first(Y) = { ε, (}

- Si a es un terminal
  entonces first(a) = { a }

- Si $\mathbf{A} \rightarrow \boldsymbol{\varepsilon}$ es una producción
  o $\mathbf{A} \rightarrow \mathbf{X_1\, X_2 \ldots\, X_k}$ es una producción
  y $\varepsilon \in$ first(X_1), …, first(X_k)
  entonces first(A) = first(A) ∪ { ε }

- Si $\mathbf{A} \rightarrow \mathbf{X_1\, X_2 \ldots\, X_i\, X_{i+1} \ldots\, X_k}$ es una producción
  y $\varepsilon \in$ first(X_1), …, first(X_i)
  y el terminal a ∈ first(X_{i+1})
  entonces first(A) = first(A) ∪ { a }

Ejemplo de first()

- Grammar

  <S> → <X> $

  <X> → <Y>

  <X> → (

  <Y> → (<Y>)

  <Y> → ε

first( )) = {) }

first(( ) = { (}

first(S) = { ε, (}

first(X) = { ε, (}

first(Y) = { ε, (}

- Si a es un terminal
  entonces first(a) = { a }

- Si **A →** **ε** es una producción
  o **A →** $\mathbf{X_1 X_2 \ldots X_k}$ es una producción
  y ε ∈ first(X_1), …, first(X_k)
  entonces first(A) = first(A) ∪ { ε }

- Si **A →** $\mathbf{X_1 X_2 \ldots X_i X_{i+1} \ldots X_k}$ es una producción
  y ε ∈ first(X_1), …, first(X_i)
  y el terminal a ∈ first(X_{i+1})
  entonces first(A) = first(A) ∪ { a }

Ejemplo de first()

- Grammar

  <S> → <X> $
  <X> → <Y>
  <X> → (
  <Y> → (<Y>)
  <Y> → ε

  first( )) = {) }
  first(( ) = { (}
  first(S) = { ε, (}
  first(X) = { ε, (}
  first(Y) = { ε, (}

- Si a es un terminal
  entonces first(a) = { a }

- Si **A → ε** es una producción
  o **A → X_1 X_2... X_k** es una producción
  y ε ∈ first(X_1), …, first(X_k)
  entonces first(A) = first(A) ∪ { ε }

- Si **A → X_1 X_2... X_i X_{i+1} ... X_k** es una producción
  y ε ∈ first(X_1), …, first(X_i)
  y el terminal a ∈ first(X_{i+1})
  entonces first(A) = first(A) ∪ { a }

Algoritmo para follow()

- follow(S) = { \$ }   donde S es el símbolo de inicio

- Si $\mathbf{A \rightarrow \alpha B \beta}$ es una producción
  entonces follow(B) = follow(B) $\cup$ (first(β) - {ε})

- Si $\mathbf{A \rightarrow \alpha B}$ es una producción
  o $\mathbf{A \rightarrow \alpha B \beta}$ es una producción y $\varepsilon \in$ first(β)
  entonces follow(B) = follow(B) $\cup$ follow(A)

Ejemplo de follow()

- Gramática

<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε

Ejemplo de follow()

- Gramática

  <S> → <X> $

  <X> → <Y>

  <X> → (

  <Y> → (<Y>)

  <Y> → ε

first( )) = {) }

first(( ) = { (}

first(S) = { (, ε }

first(X) = { (, ε }

first(Y) = { (, ε }

Ejemplo de follow()

- Gramática

<S> → <X> $
<X> → <Y>
<X> → (
<Y> → (<Y>)
<Y> → ε

first( )) = {) }
first(( ) = { (}
first(S) = { (, ε }
first(X) = { (, ε }
first(Y) = { (, ε }

follow(S) = { }
follow(X) = { }
follow(Y) = { }

Ejemplo de follow()

- Gramática

  <S> → <X> $
  <X> → <Y>
  <X> → (
  <Y> → (<Y>)
  <Y> → ε

first( )) = {) }
first(( ) = { (}
first(S) = { (, ε }
first(X) = { (, ε }
first(Y) = { (, ε }

follow(S) = { }
follow(X) = { }
follow(Y) = { }

- follow(S) = { $ } donde S es el símbolo de inicio
- Si **A → αBβ** es una producción entonces follow(B) = follow(B) ∪ (first(β) - {ε})
- Si **A → αB** es una producción
  o **A → αBβ** es una producción
  y ε ∈ first(β)
  entonces follow(B) = follow(B) ∪ follow(A)

Ejemplo de follow()

- Gramática

  <S> → <X> $
  <X> → <Y>
  <X> → (
  <Y> → (<Y>)
  <Y> → ε

first( )) = {) }
first(( ) = { (}
first(S) = { (, ε }
first(X) = { (, ε }
first(Y) = { (, ε }

- follow(S) = { $ } donde S es el símbolo de inicio
- Si **A → αBβ** es una producción entonces
  follow(B) = follow(B) ∪ (first(β) - {ε})
- Si **A → αB** es una producción
  o **A → αBβ** es una producción
  y ε ∈ first(β)
  entonces follow(B) = follow(B) ∪ follow(A)

follow(S) = { $ }
follow(X) = { }
follow(Y) = { }

Ejemplo de follow()

- Gramática

  <S> → <X> $

  <X> → <Y>

  <X> → (

  <Y> → (<Y>)

  <Y> → ε

first( )) = {) }

first(( ) = { (}

first(S) = { (, ε }

first(X) = { (, ε }

first(Y) = { (, ε }

- follow(S) = { $ } donde S es el símbolo de inicio
- Si **A → αBβ** es una producción entonces follow(B) = follow(B) ∪ (first(β) - {ε})
- Si **A → αB** es una producción o **A → αBβ** es una producción y ε ∈ first(β) entonces follow(B) = follow(B) ∪ follow(A)

follow(S) = { $ }

follow(X) = { }

follow(Y) = { }

Ejemplo de follow()

- Gramática
  <S> → <X> $
  <X> → <Y>
  <X> → (
  <Y> → (<Y>)
  <Y> → ε

first( )) = {) }
first(( ) = { (}
first(S) = { (, ε }
first(X) = { (, ε }
first(Y) = { (, ε }

- follow(S) = { $ } donde S es el símbolo de inicio
- Si **A → αBβ** es una producción entonces follow(B) = follow(B) ∪ (first(β) - {ε})
- Si **A → αB** es una producción
  o **A → αBβ** es una producción
  y ε ∈ first(β)
  entonces follow(B) = follow(B) ∪ follow(A)

follow(S) = { $ }
follow(X) = { }
follow(Y) = {) }

Ejemplo de follow()

- Gramática

  <S> → <X> $

  <X> → <Y>

  <X> → (

  <Y> → (<Y>)

  <Y> → ε

first( )) = {) }

first(( ) = { (}

first(S) = { (, ε }

first(X) = { (, ε }

first(Y) = { (, ε }

follow(S) = { $ }

follow(X) = { }

follow(Y) = {) }

- follow(S) = { $ } donde S es el símbolo de inicio
- Si **A → αBβ** es una producción entonces follow(B) = follow(B) ∪ (first(β) - {ε})
- Si **A → αB** es una producción
  o **A → αBβ** es una producción
  y ε ∈ first(β)
  entonces follow(B) = follow(B) ∪ follow(A)

Ejemplo de follow()

- Gramática

  <S> → <X> $
  <X> → <Y>
  <X> → (
  <Y> → (<Y>)
  <Y> → ε

first( )) = {) }
first(( ) = { (}
first(S) = { (, ε }
first(X) = { (, ε }
first(Y) = { (, ε }

- follow(S) = { $ } donde S es el símbolo de inicio
- Si **A → αBβ** es una producción entonces follow(B) = follow(B) ∪ (first(β) - {ε})
- Si **A → αB** es una producción o **A → αBβ** es una producción y ε ∈ first(β) entonces follow(B) = follow(B) ∪ follow(A)

follow(S) = { $ }
follow(X) = { $ }
follow(Y) = {) }

Ejemplo de follow()

- Gramática

  <S> → <X> $
  <X> → <Y>
  <X> → (
  <Y> → (<Y>)
  <Y> → ε

first( )) = {) }
first(( ) = { (}
first(S) = { (, ε }
first(X) = { (, ε }
first(Y) = { (, ε }

follow(S) = { $ }
follow(X) = { $ }
follow(Y) = {) }

- follow(S) = { $ } donde S es el símbolo de inicio
- Si **A → αBβ** es una producción entonces follow(B) = follow(B) ∪ (first(β) - {ε})
- Si **A → αB** es una producción
  o **A → αBβ** es una producción
  y ε ∈ first(β)
  entonces follow(B) = follow(B) ∪ follow(A)

Ejemplo de follow()

- Gramática

  <S> → <X> $

  <X> → <Y>

  <X> → (

  <Y> → (<Y>)

  <Y> → ε

first( )) = {) }

first(( ) = { (}

first(S) = { (, ε }

first(X) = { (, ε }

first(Y) = { (, ε }

- follow(S) = { $ } donde S es el símbolo de inicio
- Si **A → αBβ** es una producción entonces follow(B) = follow(B) ∪ (first(β) - {ε})
- Si **A → αB** es una producción o **A → αBβ** es una producción y ε ∈ first(β) entonces follow(B) = follow(B) ∪ follow(A)

follow(S) = { $ }

follow(X) = { $ }

follow(Y) = {), $ }

Ejemplo de follow()

- Gramática

  <S> → <X> $
  <X> → <Y>
  <X> → (
  <Y> → (<Y>)
  <Y> → ε

first( )) = {) }
first(( ) = { (}
first(S) = { (, ε }
first(X) = { (, ε }
first(Y) = { (, ε }

- follow(S) = { $ } donde S es el símbolo de inicio
- Si **A → αBβ** es una producción entonces follow(B) = follow(B) ∪ (first(β) - {ε})
- Si **A → αB** es una producción
  o **A → αBβ** es una producción
  y ε ∈ first(β)
  entonces follow(B) = follow(B) ∪ follow(A)

follow(S) = { $ }
follow(X) = { $ }
follow(Y) = {), $ }

Creando un Parser Engine LR(1)

- Necesitamos definir funciones closure() y goto() para ítems LR(1)
- Necesitamos proveer un algoritmo para crear el DFA
- Necesitamos proveer un algoritmo para crear la tabla de parseo

Algoritmo para Closure

colsure(I)

Algoritmo para Closure

colsure(I)

repetir

hasta que **I** no cambie

Algoritmo para Closure

colsure(I)

repetir

para todos los ítems [$\mathbf{A \rightarrow \alpha \bullet X \beta \quad c}$] en **I**

hasta que **I** no cambie

Algoritmo para Closure

colsure(I)

repetir

para todos los ítems [**A** → **α • X β   c**] en **I**

para cualquier producción **X** → **γ**

hasta que **I** no cambie

Algoritmo para Closure

colsure(I)

repetir

para todos los ítems [$\mathbf{A} \rightarrow \boldsymbol{\alpha} \bullet \mathbf{X}\, \boldsymbol{\beta}\quad \mathbf{c}$] en **I**

para cualquier producción $\mathbf{X} \rightarrow \boldsymbol{\gamma}$

para cualquier d $\in$first($\boldsymbol{\beta}$**c**)

hasta que **I** no cambie

Algoritmo para Closure

colsure(I)

repetir

para todos los ítems [$\mathbf{A} \rightarrow \boldsymbol{\alpha} \bullet \mathbf{X}\ \boldsymbol{\beta}\quad \mathbf{c}$] en **I**

para cualquier producción $\mathbf{X} \rightarrow \boldsymbol{\gamma}$

para cualquier d $\in$ first($\boldsymbol{\beta}\mathbf{c}$)

$\mathbf{I} = \mathbf{I} \cup \{\ [\mathbf{X} \rightarrow \bullet\ \boldsymbol{\gamma}\quad \mathbf{d}]\ \}$

hasta que **I** no cambie

Algoritmo para Closure

colsure(I)

repetir

para todos los ítems [**A** → **α • X β** **c**] en **I**

para cualquier producción **X** → **γ**

para cualquier d ∈first(**βc**)

I = **I** ∪ { [**X** → **• γ** **d**] }

hasta que **I** no cambie

retornar **I**

Ejemplo de Closure

- Qué es **closure([Y → (• Y) $])**

Ejemplo de Closure

- Qué es **closure([Y → (• Y) $])**

- Gramática

<S> → <X> $
<X> → <Y>
<X> → (
<Y> → (<Y>)
<Y> → ε

Ejemplo de Closure

- Qué es **closure([Y → (• Y) $])**

$$\mathbf{I} = \left\{ \begin{array}{l} [\,Y \rightarrow \ (\, \bullet Y \ )\ \ \$\,] \\ \\ \\ \end{array} \right\}$$

- Gramática

<S> → <X> $
<X> → <Y>
<X> → (
<Y> → (<Y>)
<Y> → ε

Ejemplo de Closure

- Qué es **closure**([**Y** → (• **Y**) **$**])

$$\mathbf{I} = \left\{ \begin{array}{l} [\,\mathbf{Y} \rightarrow (\,\bullet\,\mathbf{Y}\ )\ \ \mathbf{\$}\,] \\ \\ \\ \end{array} \right\}$$

repetir
  para todos los ítems [**A** → **α** • **X** **β** **c**] en **I**
    para cualquier producción **X** → **γ**
      para cualquier d ∈first(**βc**)
        I = **I** ∪ { [**X** → • **γ** **d**] }
hasta que **I** no cambie

- Gramática
  <S> → **<X>** **$**
  <X> → **<Y>**
  <X> → **(**
  <Y> → **(** **<Y>** **)**
  <Y> → **ε**

Ejemplo de Closure

- Qué es **closure**([Y → (• Y) $])

I = { **[Y → (• Y) $]** }

- Gramática

<S> → <X> $
<X> → <Y>
<X> → (
<Y> → (<Y>)
<Y> → ε

repetir
  para todos los ítems [**A** → **α** • **X** β **c**] en **I**
    para cualquier producción **X** → **γ**
      para cualquier d ∈first(**βc**)
        I = **I** ∪ { [**X** → • **γ** **d**] }
hasta que **I** no cambie

Ejemplo de Closure

- Qué es **closure**([Y → (• Y) $])

$$\mathbf{I} = \left\{ \begin{array}{l} [\ \mathbf{Y} \rightarrow\ (\ \bullet\ \mathbf{Y}\ \ )\ \ \$\ ] \\ \\ \\ \end{array} \right\}$$

repetir
para todos los ítems [**A** → **α • X β** **c**] en **I**
para cualquier producción **X** → **γ**
para cualquier d ∈first(**βc**)
I = **I** ∪ { [**X** → **• γ** **d**] }
hasta que **I** no cambie

- Gramática
  - **<S> → <X> $**
  - **<X> → <Y>**
  - **<X> → (**
  - **<Y> → (<Y>)**
  - **<Y> → ε**

Ejemplo de Closure

- Qué es **closure**([Y → (• Y) $])

I = { **[Y → (• Y) $]** }

First( )$) = ???

- Gramática

<S> → <X> $
<X> → <Y>
<X> → (
<Y> → (<Y>)
<Y> → ε

repetir
para todos los ítems [**A** → **α** • **X β c**] en **I**
para cualquier producción **X** → **γ**
para cualquier d ∈first(**βc**)
I = **I** ∪ { [**X** → • **γ d**] }
hasta que **I** no cambie

Ejemplo de Closure

- Qué es **closure**([Y → (• Y) $])

I = { [**Y** → **(• Y)** **$**] }

First( )$) = {) }

repetir
para todos los ítems [**A** → **α** • **X β** **c**] en **I**
para cualquier producción **X** → **γ**
para cualquier d ∈first(**βc**)
I = **I** ∪ { [**X** → • **γ** **d**] }
hasta que **I** no cambie

- Gramática

<S> → **<X> $**
<X> → **<Y>**
<X> → **(**
<Y> → **(<Y>)**
<Y> → **ε**

Ejemplo de Closure

- Qué es **closure**([$Y \rightarrow (\bullet Y)\ \$$])

$$I = \left\{ \begin{array}{l} [\,Y \rightarrow (\bullet Y\ )\ \ \$\,] \\ [\,Y \rightarrow \bullet(\ \ Y\ \ )\ \ )\,] \\ \end{array} \right\}$$

$$\mathbf{First}(\ )\$\ ) = \{\ )\ \}$$

repetir

para todos los ítems $[A \rightarrow \alpha \bullet X \beta\ \ c]$ en $\mathbf{I}$

para cualquier producción $X \rightarrow \gamma$

para cualquier d $\in$first(βc)

$\mathbf{I} = \mathbf{I} \cup \{\ [X \rightarrow \bullet \gamma\ \ d]\ \}$

hasta que **I** no cambie

- Gramática

$\langle S\rangle \rightarrow \langle X\rangle\ \$$

$\langle X\rangle \rightarrow \langle Y\rangle$

$\langle X\rangle \rightarrow ($

$\langle Y\rangle \rightarrow (\ \langle Y\rangle\ )$

$\langle Y\rangle \rightarrow \varepsilon$

Ejemplo de Closure

- Qué es **closure**([Y → (• Y) $])

$$\mathbf{I} = \left\{ \begin{array}{l} [\, Y \rightarrow (\, \bullet Y \;\;) \;\; \$ \,] \\ [\, Y \rightarrow \bullet (\;\; Y \;\;) \;\;) \,] \\ \\ \end{array} \right\}$$

- Gramática

$\langle S \rangle \rightarrow \langle X \rangle \; \$$

$\langle X \rangle \rightarrow \langle Y \rangle$

$\langle X \rangle \rightarrow ($

$\langle Y \rangle \rightarrow (\; \langle Y \rangle \;)$

$\langle Y \rangle \rightarrow \varepsilon$

repetir
  para todos los ítems [**A** → **α** • **X** **β** **c**] en **I**
    para cualquier producción **X** → **γ**
      para cualquier d ∈first(**βc**)
        I = **I** ∪ { [**X** → • **γ** **d**] }
hasta que **I** no cambie

Ejemplo de Closure

- Qué es **closure**([Y → (• Y) $])

$$\mathbf{I} = \left\{ \begin{array}{l} [\ \mathbf{Y} \rightarrow\ (\ \bullet\ \mathbf{Y}\ \ )\ \ \$\ ] \\ [\ \mathrm{Y} \rightarrow \bullet\ (\ \ \mathrm{Y}\ \ )\ \ )\ ] \\ \\ \end{array} \right\}$$

repetir
para todos los ítems [A → α • X β c] en **I**
para cualquier producción **X** → γ
para cualquier d ∈first(**βc**)
I = **I** ∪ { [**X** → • γ **d**] }
hasta que **I** no cambie

- Gramática
  - **<S>** → **<X>** $
  - **<X>** → **<Y>**
  - **<X>** → (
  - **<Y>** → (**<Y>**)
  - **<Y>** → ε

Ejemplo de Closure

- Qué es **closure**([Y → (• Y) $])

$$\mathbf{I} = \left\{ \begin{array}{l} [\ Y \rightarrow (\ \bullet\ Y\ )\ \ \$\ ] \\ [\ Y \rightarrow \bullet\ (\ Y\ )\ \ )\ ] \\ \\ \end{array} \right\}$$

First( )$) = {) }

- Gramática

<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε

repetir

para todos los ítems [**A** → **α** • **X β** **c**] en **I**

para cualquier producción **X** → **γ**

para cualquier d ∈first(**βc**)

I = **I** ∪ { [**X** → • **γ** **d**] }

hasta que **I** no cambie

Ejemplo de Closure

- Qué es **closure**([**Y** → (• **Y**) **$**])

$$\mathbf{I} = \left\{ \begin{array}{l} [\,\mathbf{Y} \rightarrow \ (\ \bullet\ \mathbf{Y}\ \ )\ \ \mathbf{\$}\,] \\ [\,\mathbf{Y} \rightarrow \bullet\ (\ \ \mathbf{Y}\ \ )\ \ )\,] \\ [\,\mathbf{Y} \rightarrow \bullet \qquad\qquad\quad)\,] \end{array} \right\}$$

$$\mathbf{First}(\ )\$\ ) = \{\ )\ \}$$

- Gramática

<S> → <X> $
<X> → <Y>
<X> → (
<Y> → (<Y>)
<Y> → ε

repetir
  para todos los ítems [**A** → **α** • **X** **β** **c**] en **I**
    para cualquier producción **X** → **γ**
      para cualquier d ∈first(**βc**)
        I = **I** ∪ { [**X** → • **γ** **d**] }
hasta que **I** no cambie

Ejemplo de Closure

- Qué es **closure**([**Y** → (• **Y**) $])

$$\mathbf{I} = \left\{ \begin{array}{l} [\ \mathbf{Y} \rightarrow\ (\ \bullet\ \mathbf{Y}\ \ )\ \ \$\ ] \\ [\ \mathbf{Y} \rightarrow \bullet\ (\ \ \mathbf{Y}\ \ )\ \ )\ ] \\ [\ \mathbf{Y} \rightarrow \bullet\ \ \ \ \ \ \ \ \ \ \ \ \ )\ ] \end{array} \right\}$$

repetir
  para todos los ítems [**A** → **α** • **X** **β**   **c**] en **I**
    para cualquier producción **X** → **γ**
      para cualquier d ∈first(**βc**)
        I = **I** ∪ { [**X** → • **γ**   **d**] }
hasta que **I** no cambie

- Gramática
  <S> → **<X> $**
  <X> → **<Y>**
  <X> → (
  <Y> → (**<Y>**)
  <Y> → **ε**

Ejemplo de Closure

- Qué es **closure**([Y → (• Y) $])

$$\mathbf{I} = \left\{ \begin{array}{l} [\, Y \rightarrow (\, \bullet\, Y \;)\;\; \$\,] \\ [\, Y \rightarrow \bullet\, (\;\; Y \;)\;\;)\,] \\ [\, Y \rightarrow \bullet \qquad\qquad\quad)\,] \end{array} \right\}$$

repetir
  para todos los ítems [**A** → **α** • **X β**   **c**] en **I**
    para cualquier producción **X** → **γ**
      para cualquier d ∈first(**βc**)
        I = **I** ∪ { [**X** → • **γ**   **d**] }
hasta que **I** no cambie

- Gramática
  <S> → **<X> $**
  <X> → **<Y>**
  <X> → **(**
  <Y> → **(<Y>)**
  <Y> → **ε**

Ejemplo de Closure

- Qué es **closure**$([\, \mathbf{Y} \rightarrow (\bullet \mathbf{Y})\ \ \$\,])$

$$\mathbf{I} = \left\{ \begin{array}{l} [\, \mathbf{Y} \rightarrow (\bullet \mathbf{Y})\ \ \$\,] \\ [\, \mathbf{Y} \rightarrow \bullet (\ \mathbf{Y}\ )\ \ )\,] \\ [\, \mathbf{Y} \rightarrow \bullet\ \ )\,] \end{array} \right\}$$

- Gramática

$\langle \mathbf{S} \rangle \rightarrow \langle \mathbf{X} \rangle\ \$$

$\langle \mathbf{X} \rangle \rightarrow \langle \mathbf{Y} \rangle$

$\langle \mathbf{X} \rangle \rightarrow ($

$\langle \mathbf{Y} \rangle \rightarrow (\ \langle \mathbf{Y} \rangle\ )$

$\langle \mathbf{Y} \rangle \rightarrow \varepsilon$

repetir
  para todos los ítems $[\mathbf{A} \rightarrow \alpha \bullet \mathbf{X}\, \beta\ \ \mathbf{c}]$ en $\mathbf{I}$
    para cualquier producción $\mathbf{X} \rightarrow \gamma$
      para cualquier d $\in$first(β**c**)
        $\mathbf{I} = \mathbf{I} \cup \{\ [\mathbf{X} \rightarrow \bullet\, \gamma\ \ \mathbf{d}]\ \}$
hasta que **I** no cambie

Pregunta: Encontrar Closure

- Qué es **closure**([**X** → • **Y** $])

$$\mathbf{I} = \left\{ \quad ???? \quad \right\}$$

repetir
  para todos los ítems [**A** → **α** • **X β** **c**] en **I**
    para cualquier producción **X** → **γ**
      para cualquier d ∈first(**βc**)
        I = **I** ∪ { [**X** → • **γ** **d**] }
hasta que **I** no cambie

- Gramática
  <S> → **<X> $**
  <X> → **<Y>**
  <X> → **(**
  <Y> → **(<Y>)**
  <Y> → **ε**

Pregunta: Encontrar Closure

- Qué es **closure**([$X \rightarrow \bullet Y \quad \$$])

$$\mathbf{I} = \left\{ \begin{array}{l} [\, X \rightarrow \bullet Y \qquad\qquad \$ \,] \\ [\, Y \rightarrow \bullet (\quad Y \quad) \quad \$ \,] \\ [\, Y \rightarrow \bullet \qquad\qquad\qquad \$ \,] \end{array} \right\}$$

repetir

para todos los ítems [$\mathbf{A} \rightarrow \alpha \bullet \mathbf{X}\, \beta \quad \mathbf{c}$] en **I**

para cualquier producción $\mathbf{X} \rightarrow \gamma$

para cualquier d $\in$first(β**c**)

$\mathbf{I} = \mathbf{I} \cup \{\ [\mathbf{X} \rightarrow \bullet\, \gamma \quad \mathbf{d}]\ \}$

hasta que **I** no cambie

- Gramática

$$\begin{array}{lll} \langle S \rangle & \rightarrow & \langle X \rangle\ \$ \\ \langle X \rangle & \rightarrow & \langle Y \rangle \\ \langle X \rangle & \rightarrow & (\\ \langle Y \rangle & \rightarrow & (\ \langle Y \rangle\ ) \\ \langle Y \rangle & \rightarrow & \varepsilon \end{array}$$

Algoritmo para Goto

goto(I, X)

Algoritmo para Goto

goto(I, X)

J = { }

Algoritmo para Goto

goto(I, X)

J = { }

para cualquier ítem [**A** → **α • X β** **c**] en **I**

Algoritmo para Goto

goto(I, X)

$\mathbf{J} = \{\ \}$

para cualquier ítem $[\mathbf{A} \rightarrow \boldsymbol{\alpha} \bullet \mathbf{X}\ \boldsymbol{\beta}\quad \mathbf{c}]$ en **I**

$\mathbf{J} = \mathbf{J} \cup \{[\mathbf{A} \rightarrow \boldsymbol{\alpha}\ \mathbf{X} \bullet \boldsymbol{\beta}\quad \mathbf{c}]\}$

Algoritmo para Goto

goto(I, X)

J = { }

para cualquier ítem [$\mathbf{A \rightarrow \alpha \bullet X \beta \quad c}$] en **I**

$\mathbf{J} = \mathbf{J} \cup \{[\mathbf{A \rightarrow \alpha\, X \bullet \beta \quad c}]\}$

retornar **J**

Ejemplo de Goto

- Qué es **goto([Y → (• Y) $], Y)**

Ejemplo de Goto

- Qué es **goto**([**Y** → (• **Y**) **$**], **Y**)

goto(I, X)

J = { }

para cualquier ítem [**A** → **α** • **X** **β** **c**] en **I**

J = **J** ∪ {[**A** → **α** **X** • **β** **c**]}

Ejemplo de Goto

- Qué es **goto**([**Y** → (• **Y**) **$**], **Y**)

  { }

goto(I, X)

J = { }

para cualquier ítem [**A** → **α** • **X β** **c**] en **I**

J = J ∪ {[**A** → **α X** • **β** **c**]}

Ejemplo de Goto

- Qué es **goto([Y → (• Y) $], Y)**

{ }

goto(I, X)

J = { }

para cualquier ítem [**A → α • X β c**] en **I**

J = J ∪ {[A → α X • β c]}

Ejemplo de Goto

- Qué es **goto**([**Y** → (• **Y**) **$**], **Y**)

  {[**Y** → (**Y** •) **$**]}

goto(I, X)

J = { }

para cualquier ítem [**A** → α • **X** β **c**] en **I**

J = **J** ∪ {[**A** → α **X** • β **c**]}

Ejemplo de Goto

- Qué es **goto**([**Y** → (• **Y**) **$**], **Y**)

  {[**Y** → (**Y** •) **$**]}

goto(I, X)

J = { }

para cualquier ítem [**A** → **α** • **X** **β** **c**] en **I**

J = **J** ∪ {[**A** → **α** **X** • **β** **c**]}

Construyendo el DFA

- Comenzar con el ítem [**<S'>** $\rightarrow$ • **<S>** \$?]
  - donde ? representa cualquier terminal
- Encontrar closure del ítem y crear un estado
- Elegir un estado **I**
  - para cada ítem [A$\rightarrow$ α • X β c] en **I**
    - encontrar **goto(I, X)**
    - si **goto(I, X)** no es ya un estado, creamos uno
    - agregar una arista X del estado **I** al estado **goto(I, X)**
- Repetir hasta que no sea posible agregar nada más

DFA del Ejemplo Expandido

<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε

DFA del Ejemplo Expandido

<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε

[<S> → • <X> $?]

DFA del Ejemplo Expandido

<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε

closure([**<S> → • <X> $?**]) = ???

[**<S> → • <X> $?**]

DFA del Ejemplo Expandido

<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε

closure([**<S>** → • **<X> $?**]) = ???

first($?) = ???

[**<S> → • <X> $?**]

DFA del Ejemplo Expandido

<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε

closure([**<S> → • <X> $?**]) = ???

first($?) = { $ }

[**<S> → • <X> $?**]

DFA del Ejemplo Expandido

<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε

closure([**<S>** → • **<X> $?**]) = ???

first($?) = { $ }

[**<S>** → • **<X> $?**]
[**<X>** → • **<Y> $**]
[**<X>** → • **($**]
[**<Y>** → • **(<Y>) $**]
[**<Y>** → • $]

DFA del Ejemplo Expandido

<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε

s0

[<S> → • <X> $?]
[<X> → • <Y> $]
[<X> → • ($]
[<Y> → • (<Y>) $]
[<Y> → • $]

DFA del Ejemplo Expandido

<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε

goto(, ( ) = ???

s0

[<S>	→ • <X> $	?]
[<X>	→ • <Y>	$]
[<X>	→ • (	$]
[<Y>	→ • (<Y>)	$]
[<Y>	→ •	$]

(

DFA del Ejemplo Expandido

<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε

goto(, ( ) = ???

s0

[<S> → • <X> $?]
[<X> → • <Y> $]
[<X> → • ($]
[<Y> → • (<Y>) $]
[<Y> → • $]

(

[<X> → (• $]
[<Y> → (• <Y>) $]

DFA del Ejemplo Expandido

<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε

closure( ) = ???

s0

[<S> → • <X> $?]
[<X> → • <Y> $]
[<X> → • ($]
[<Y> → • (<Y>) $]
[<Y> → • $]

(

[<X> → (• $]
[<Y> → (• <Y>) $]

DFA del Ejemplo Expandido

<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε

closure( ) = ???

s0

[<S> → • <X> $?]
[<X> → • <Y> $]
[<X> → • ($]
[<Y> → • (<Y>) $]
[<Y> → • $]

(

[<X> → (• $]
[<Y> → (• <Y>) $]
[<Y> → • (<Y>))]
[<Y> → •)]

DFA del Ejemplo Expandido

<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε

s0

[<S> → • <X> $?]
[<X> → • <Y> $]
[<X> → • ($]
[<Y> → • (<Y>) $]
[<Y> → • $]

(

s1

[<X> → (• $]
[<Y> → (• <Y>) $]
[<Y> → • (<Y>))]
[<Y> → •)]

DFA del Ejemplo Expandido

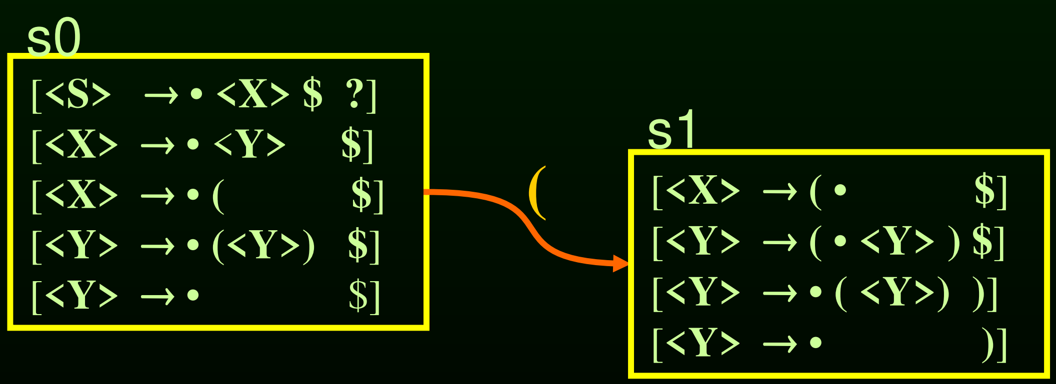
<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε



DFA del Ejemplo Expandido

$\langle S \rangle \rightarrow \langle X \rangle \ \$$

$\langle X \rangle \rightarrow \langle Y \rangle$

$\langle X \rangle \rightarrow ($

$\langle Y \rangle \rightarrow (\ \langle Y \rangle \ )$

$\langle Y \rangle \rightarrow \varepsilon$

goto(, ( ) = ???

s0

$[\langle S \rangle \rightarrow \bullet \langle X \rangle \ \$ \quad ?]$
$[\langle X \rangle \rightarrow \bullet \langle Y \rangle \quad \$]$
$[\langle X \rangle \rightarrow \bullet (\quad \$]$
$[\langle Y \rangle \rightarrow \bullet (\langle Y \rangle) \quad \$]$
$[\langle Y \rangle \rightarrow \bullet \quad \$]$

(

s1

$[\langle X \rangle \rightarrow (\bullet \quad \$]$
$[\langle Y \rangle \rightarrow (\bullet \langle Y \rangle) \ \$]$
$[\langle Y \rangle \rightarrow \bullet (\langle Y \rangle) \quad)]$
$[\langle Y \rangle \rightarrow \bullet \quad)]$

(

DFA del Ejemplo Expandido

<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε

s0

[<S> → • <X> $?]
[<X> → • <Y> $]
[<X> → • ($]
[<Y> → • (<Y>) $]
[<Y> → • $]

(

s1

[<X> → (• $]
[<Y> → (• <Y>) $]
[<Y> → • (<Y>))]
[<Y> → •)]

(

goto(, ( ) = ???

[<Y> → (• <Y>))]

DFA del Ejemplo Expandido

<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε

closure( ) = ???

[<Y> → (• <Y>))]

(

s0

[<S> → • <X> $?]
[<X> → • <Y> $]
[<X> → • ($]
[<Y> → • (<Y>) $]
[<Y> → • $]

(

s1

[<X> → (• $]
[<Y> → (• <Y>) $]
[<Y> → • (<Y>))]
[<Y> → •)]

DFA del Ejemplo Expandido

<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε

first( ))) = ???

closure( ) = ???

s0

[<S> → • <X> $?]
[<X> → • <Y> $]
[<X> → • ($]
[<Y> → • (<Y>) $]
[<Y> → • $]

(

s1

[<X> → (• $]
[<Y> → (• <Y>) $]
[<Y> → • (<Y>))]
[<Y> → •)]

(

[<Y> → (• <Y>))]

DFA del Ejemplo Expandido

$\langle S \rangle \rightarrow \langle X \rangle \ \$$

$\langle X \rangle \rightarrow \langle Y \rangle$

$\langle X \rangle \rightarrow ($

$\langle Y \rangle \rightarrow (\ \langle Y \rangle \ )$

$\langle Y \rangle \rightarrow \varepsilon$

first( ))) = {) }

closure( ) = ???

$[\langle Y \rangle \rightarrow (\bullet \langle Y \rangle) \quad)]$

(

s0

$[\langle S \rangle \rightarrow \bullet \langle X \rangle \ \$ \quad ?]$
$[\langle X \rangle \rightarrow \bullet \langle Y \rangle \quad \$]$
$[\langle X \rangle \rightarrow \bullet (\quad \$]$
$[\langle Y \rangle \rightarrow \bullet (\langle Y \rangle) \quad \$]$
$[\langle Y \rangle \rightarrow \bullet \quad \$]$

(

s1

$[\langle X \rangle \rightarrow (\bullet \quad \$]$
$[\langle Y \rangle \rightarrow (\bullet \langle Y \rangle) \ \$]$
$[\langle Y \rangle \rightarrow \bullet (\langle Y \rangle) \quad)]$
$[\langle Y \rangle \rightarrow \bullet \quad)]$

DFA del Ejemplo Expandido

<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε

first( ))) = {) }

closure( ) = ???

s0

[<S> → • <X> $?]
[<X> → • <Y> $]
[<X> → • ($]
[<Y> → • (<Y>) $]
[<Y> → • $]

(

s1

[<X> → (• $]
[<Y> → (• <Y>) $]
[<Y> → • (<Y>))]
[<Y> → •)]

(

[<Y> → (• <Y>))]
[<Y> → • (<Y>))]
[<Y> → •)]

DFA del Ejemplo Expandido

<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε

s0

[<S> → • <X> $?]
[<X> → • <Y> $]
[<X> → • ($]
[<Y> → • (<Y>) $]
[<Y> → • $]

(

s1

[<X> → (• $]
[<Y> → (• <Y>) $]
[<Y> → • (<Y>))]
[<Y> → •)]

(

s2

[<Y> → (• <Y>))]
[<Y> → • (<Y>))]
[<Y> → •)]

DFA del Ejemplo Expandido

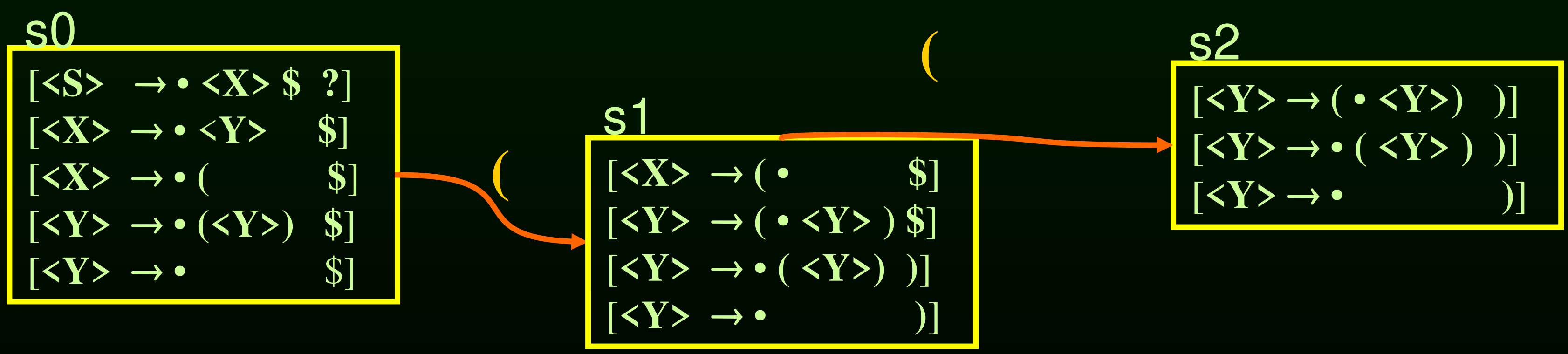
<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε



DFA del Ejemplo Expandido

<S> → <X> $

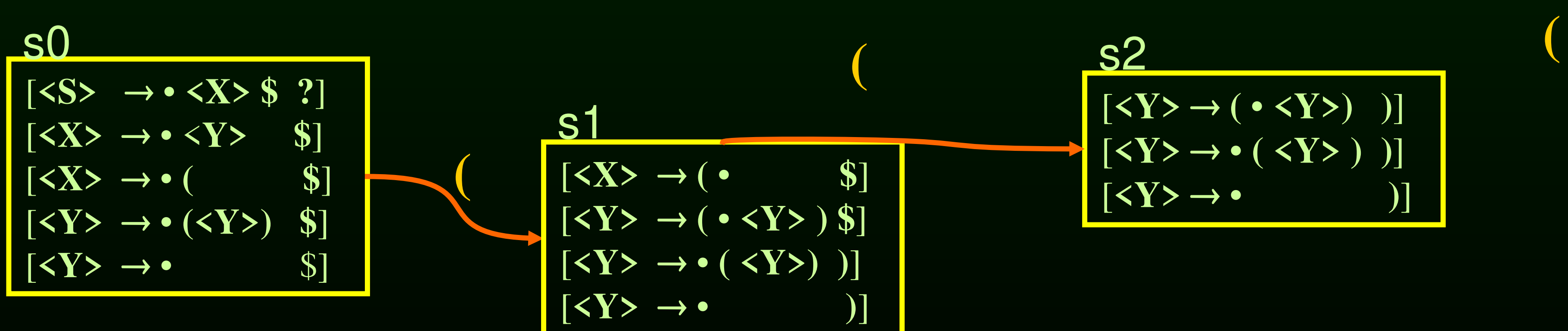
<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε

[<Y> → (• <Y>))]



DFA del Ejemplo Expandido

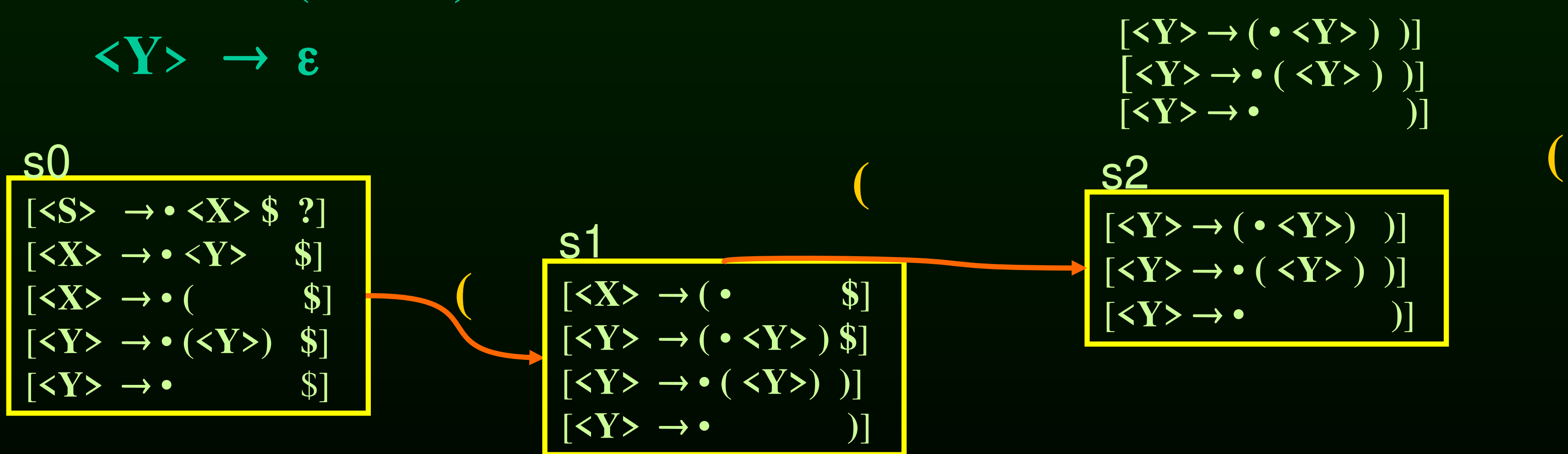
<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε



DFA del Ejemplo Expandido

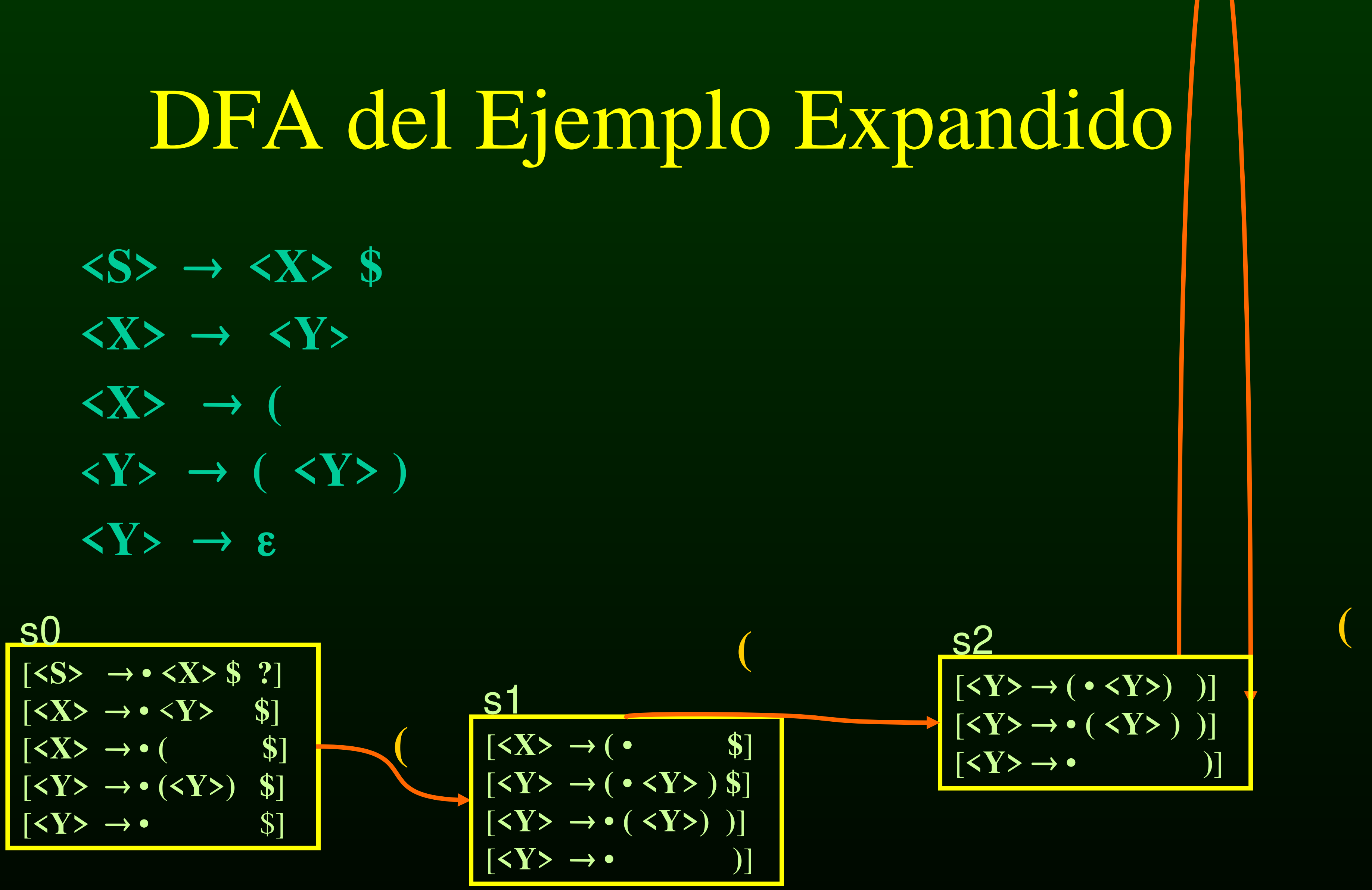
$\langle S \rangle \rightarrow \langle X \rangle \ \$$

$\langle X \rangle \rightarrow \langle Y \rangle$

$\langle X \rangle \rightarrow ($

$\langle Y \rangle \rightarrow (\ \langle Y \rangle \ )$

$\langle Y \rangle \rightarrow \varepsilon$



DFA del Ejemplo Expandido

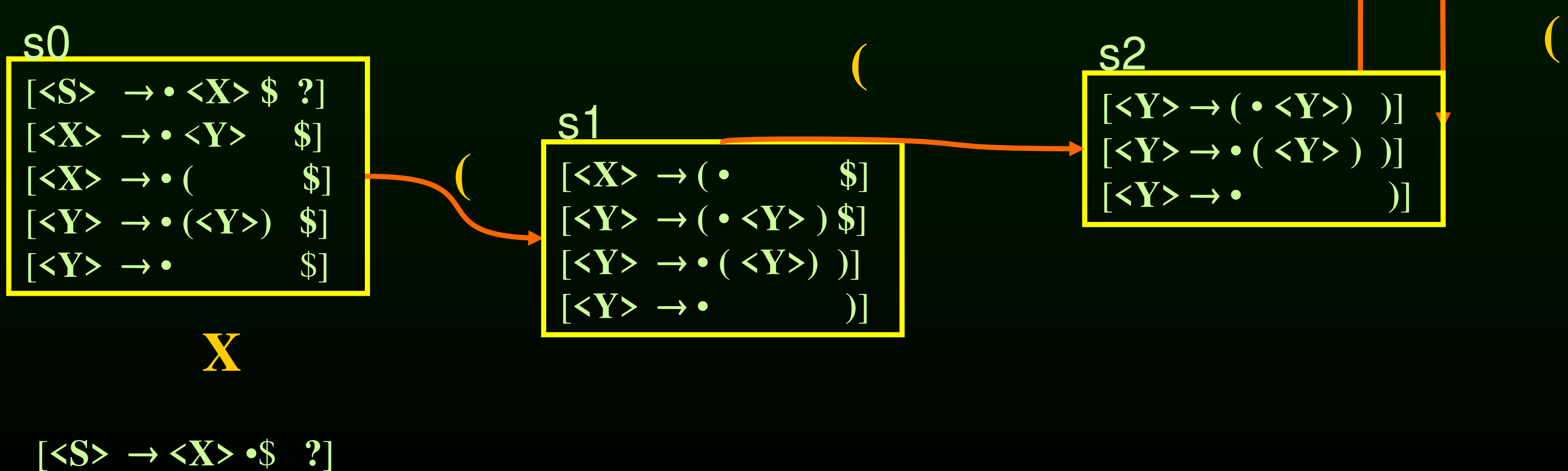
<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε



DFA del Ejemplo Expandido

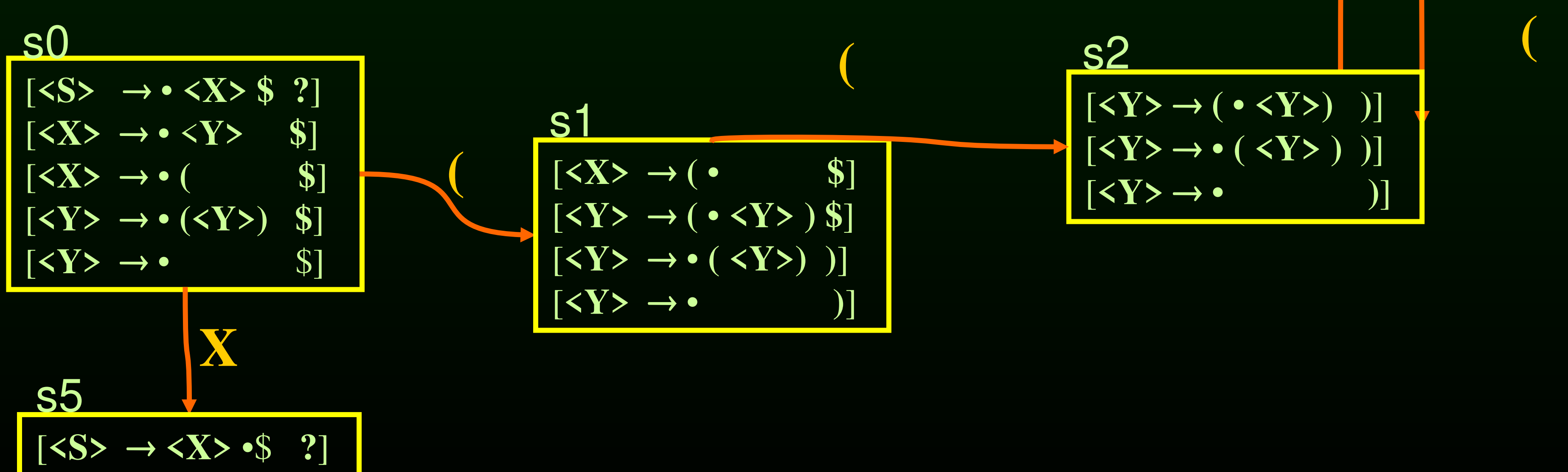
<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε



DFA del Ejemplo Expandido

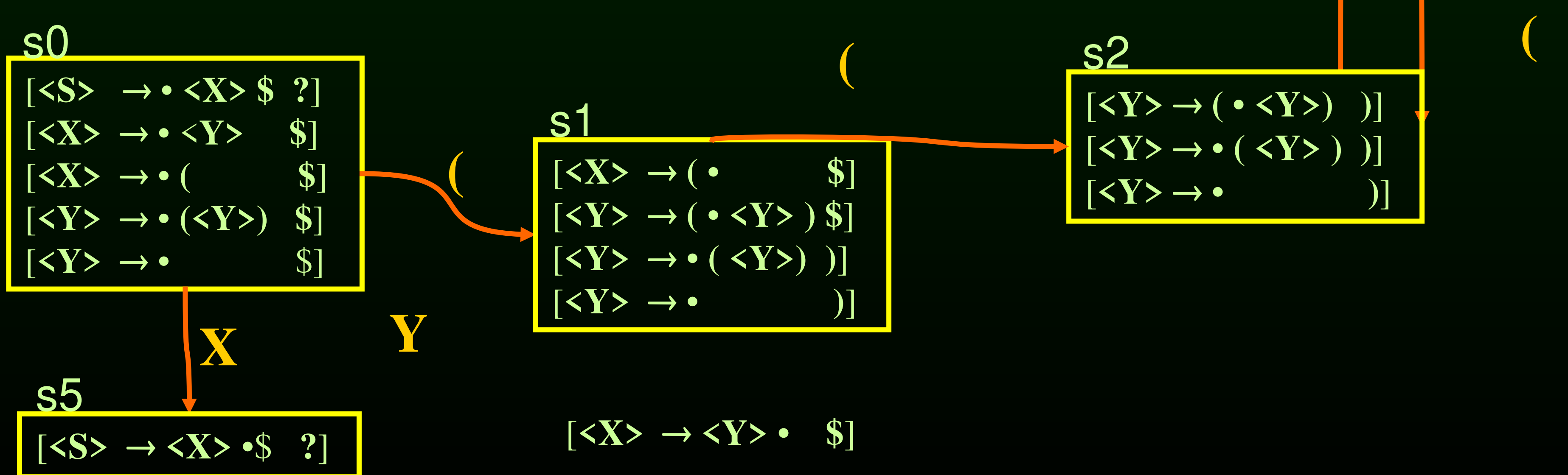
<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε



DFA del Ejemplo Expandido

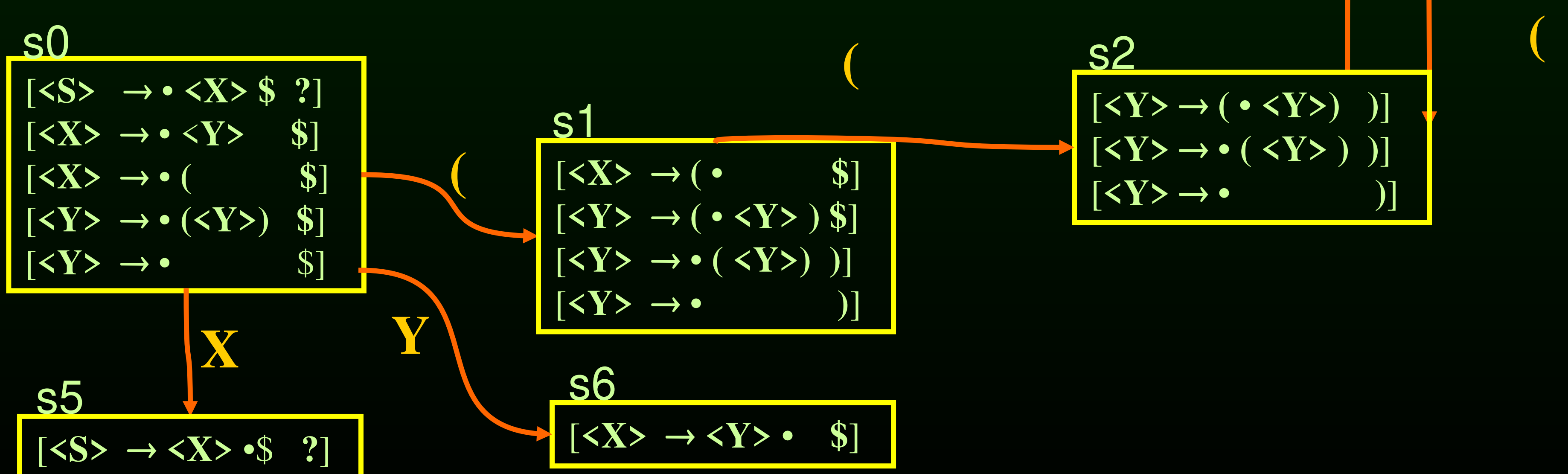
<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε



DFA del Ejemplo Expandido

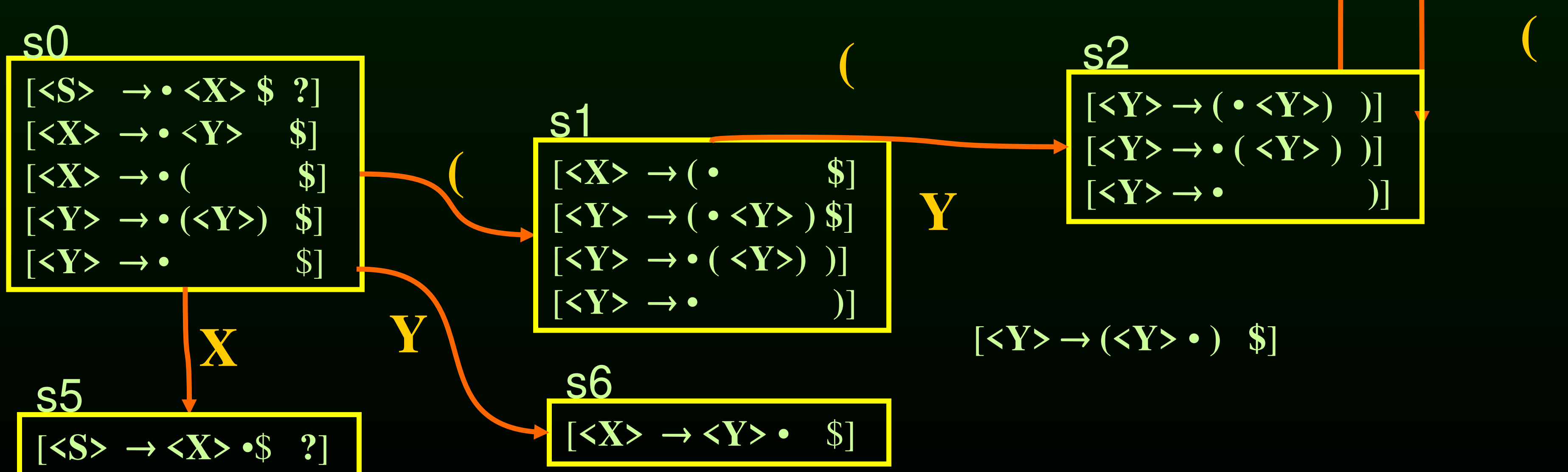
<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε



DFA del Ejemplo Expandido

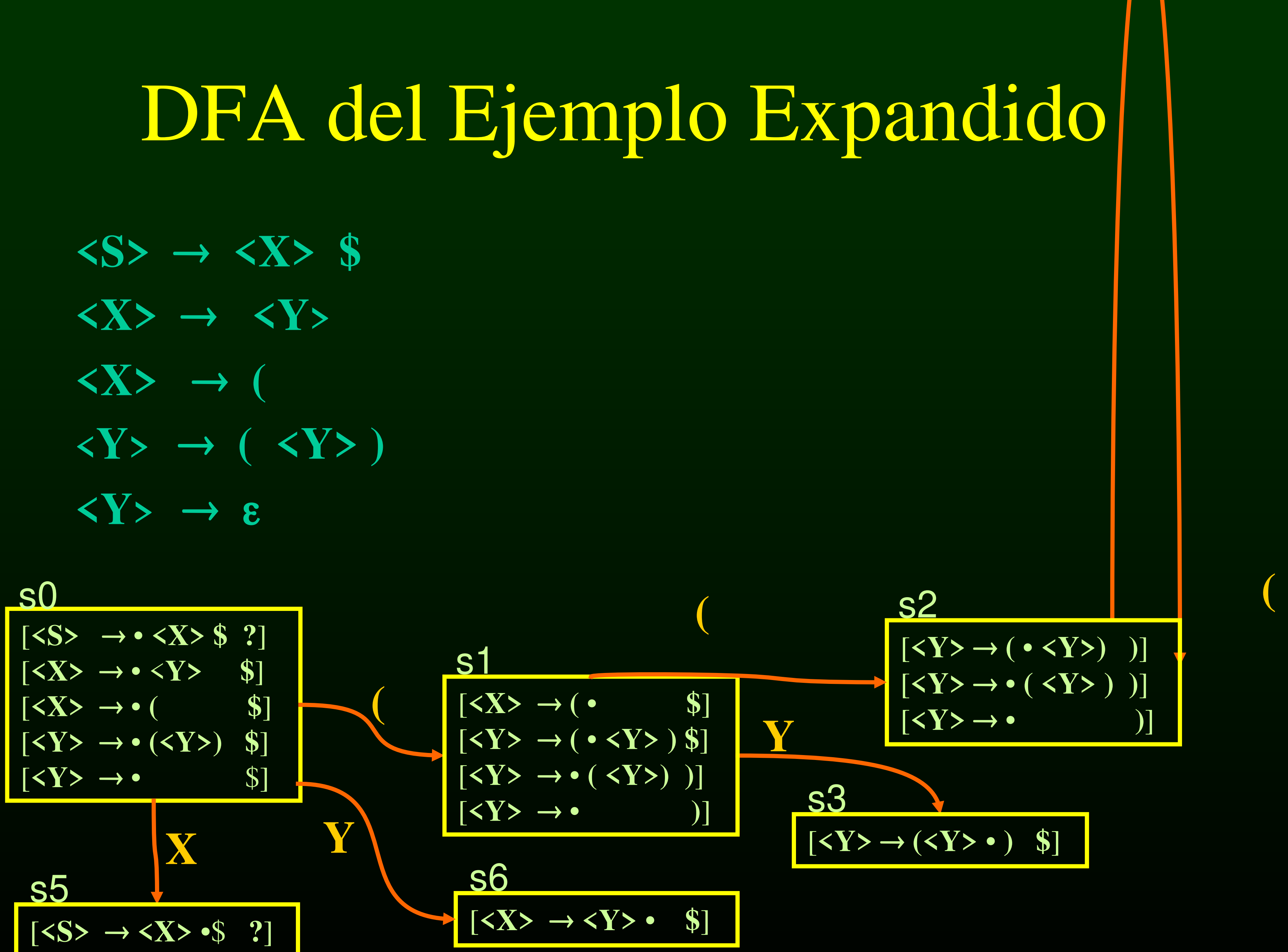
<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε



DFA del Ejemplo Expandido

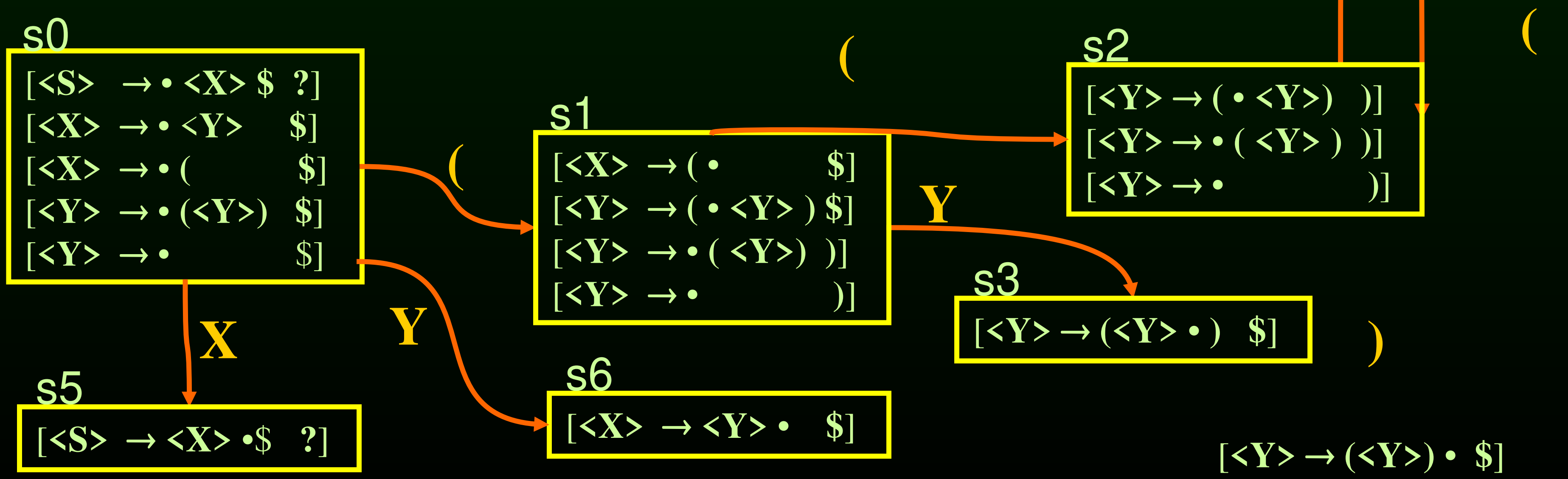
$\langle S \rangle \to \langle X \rangle \ \$$

$\langle X \rangle \to \langle Y \rangle$

$\langle X \rangle \to ($

$\langle Y \rangle \to (\ \langle Y \rangle \ )$

$\langle Y \rangle \to \varepsilon$



DFA del Ejemplo Expandido

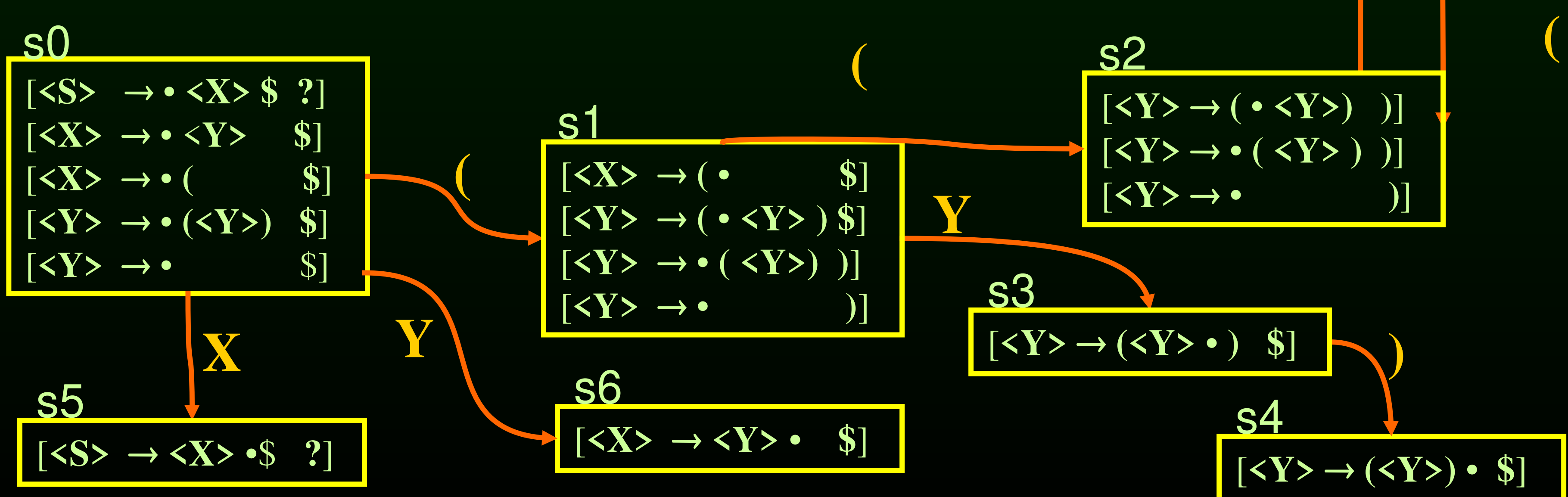
<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε



DFA del Ejemplo Expandido

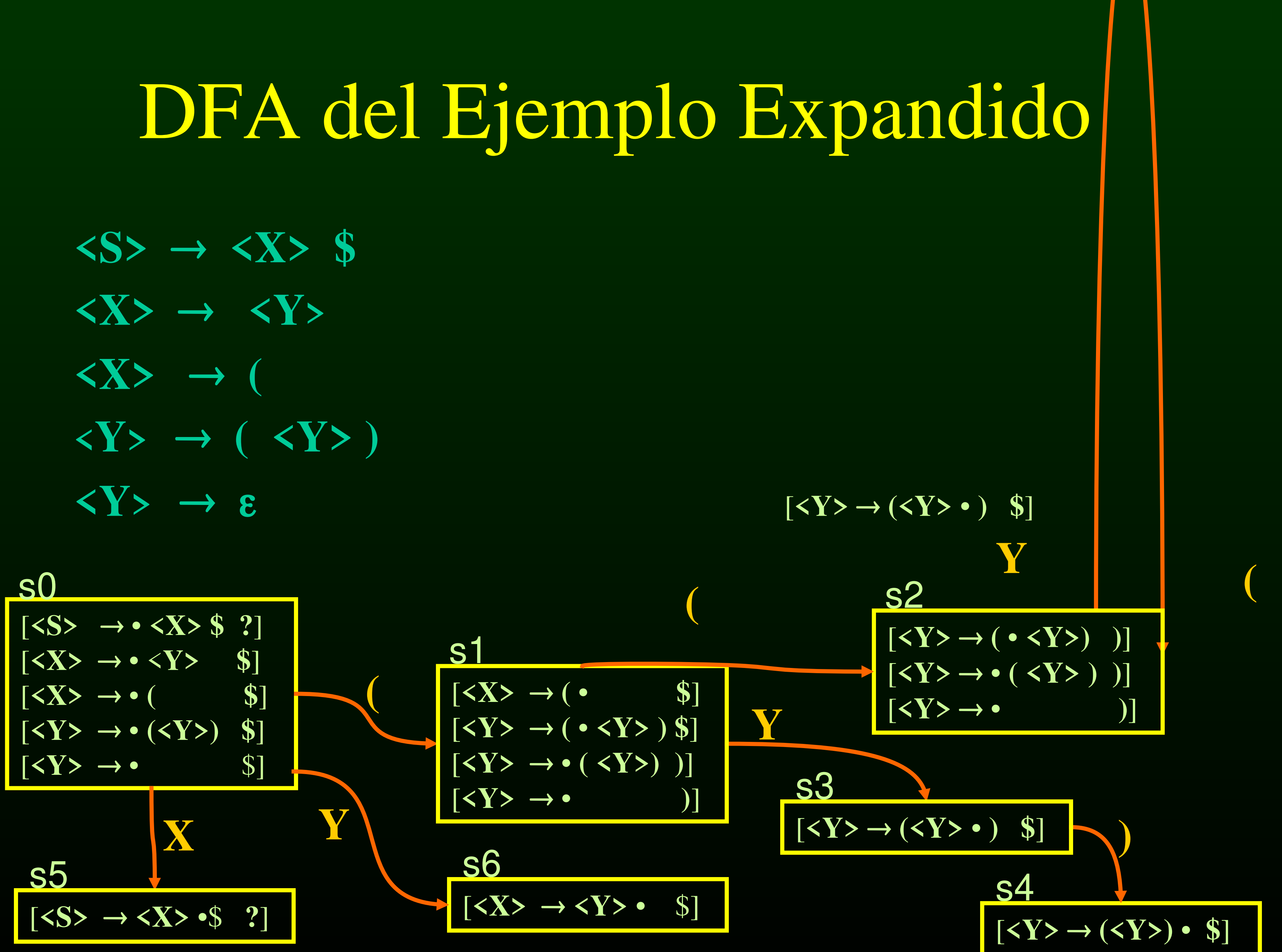
<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

<Y> → ε



DFA del Ejemplo Expandido

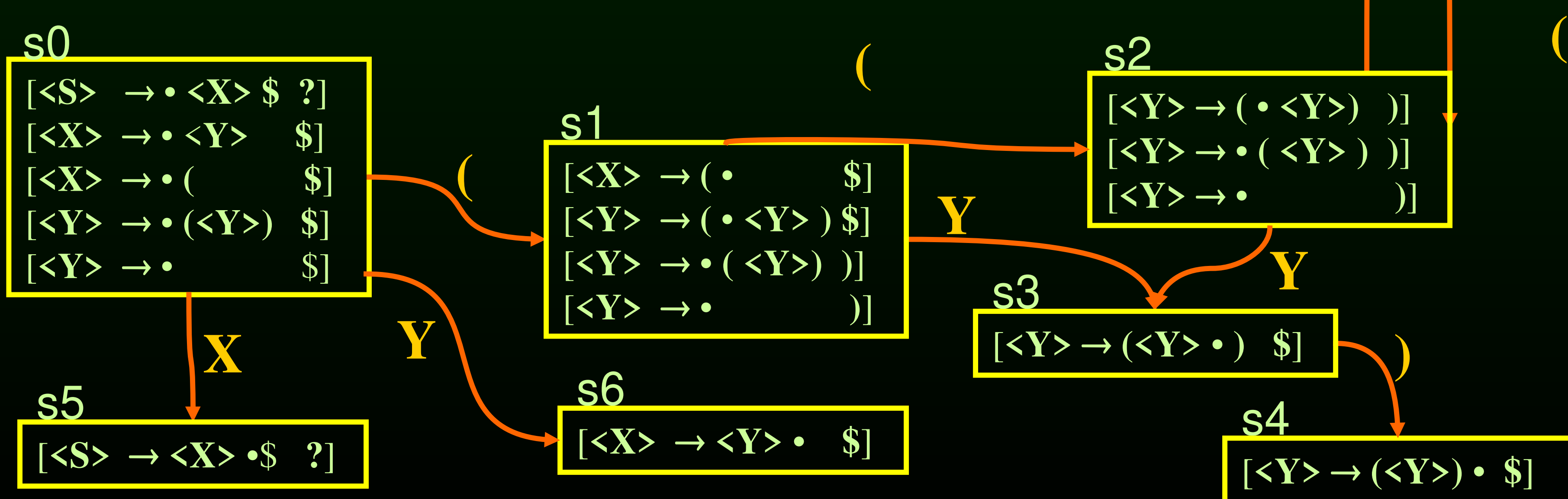
<S> → <X> $

<X> → <Y>

<X> → (

<Y> → (<Y>)

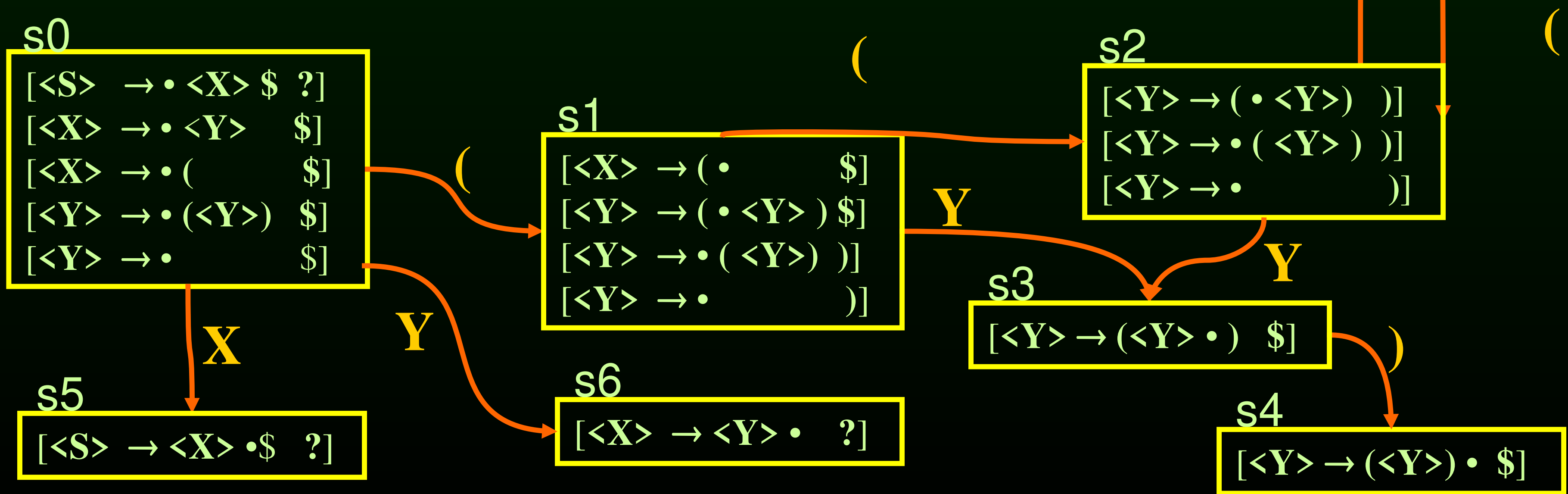
<Y> → ε



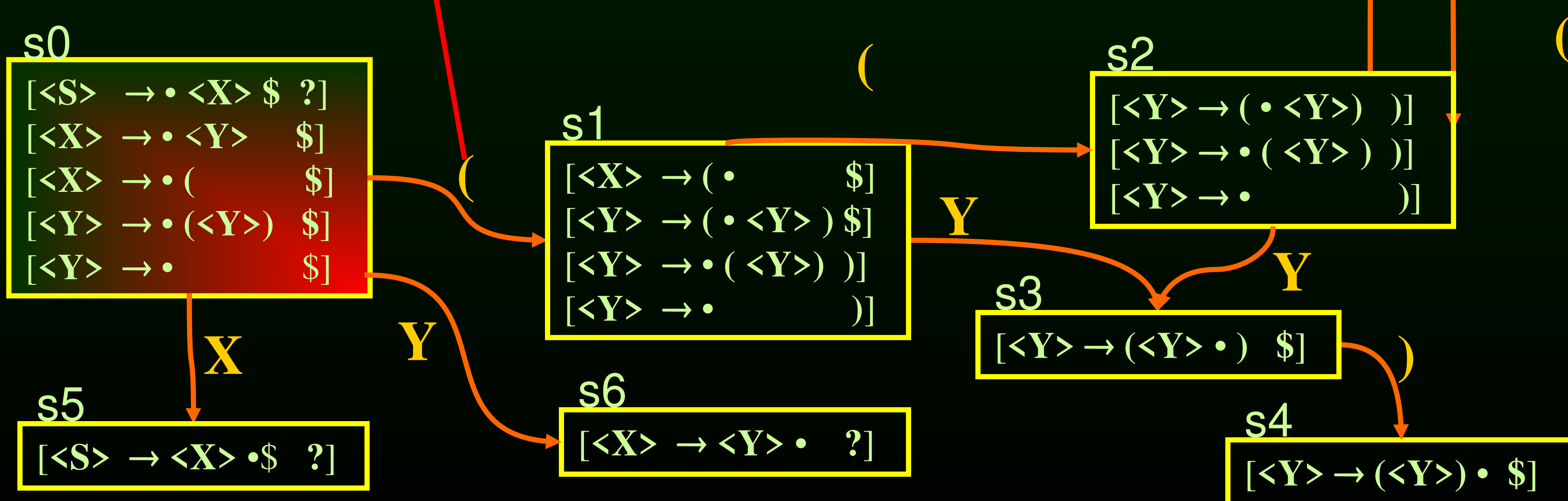
Creando las tablas de parseo

- Para cada estado
  - Transición a otro estado usando un símbolo terminal es un shift a ese estado (*shift to sn*)
  - Transición a otro estado usando un símbolo no-terminal es un goto a ese estado (*goto sn*)
  - Si hay un ítem [**A** $\rightarrow$ **α •** a] en el estado, hacemos una reducción para el símbolo de entrada "a" con la producción (*reduce k*)

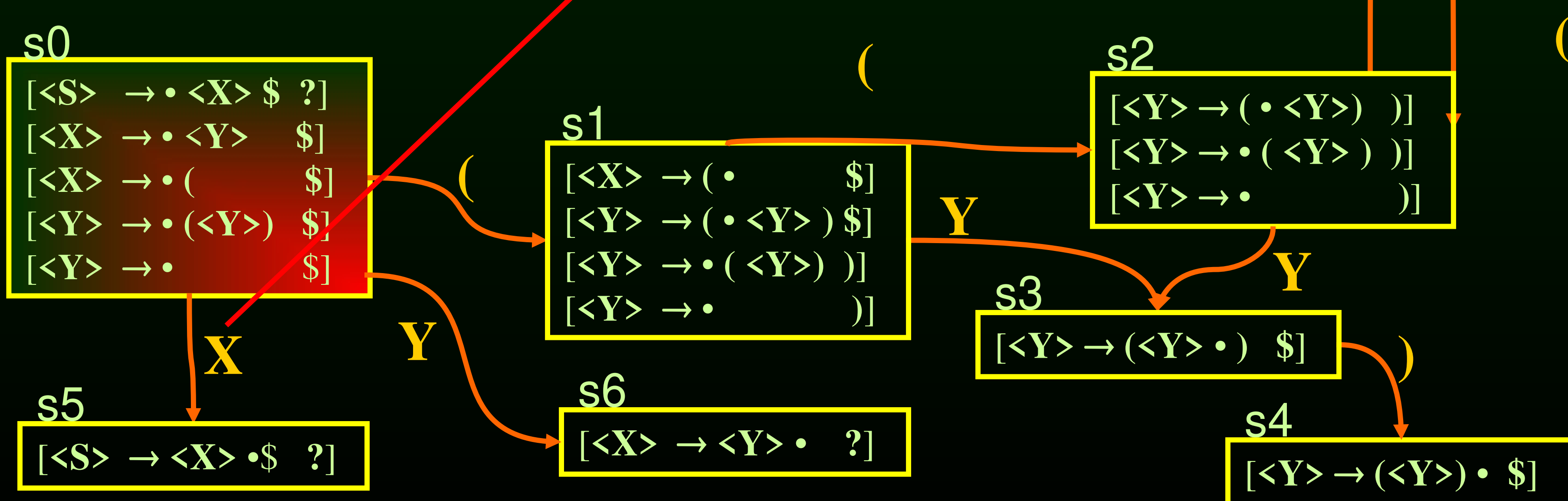
	ACTION			Goto	
State	(	)	$	X	Y
s0					
s1					
s2					
s3					
s4					
s5					
s6					



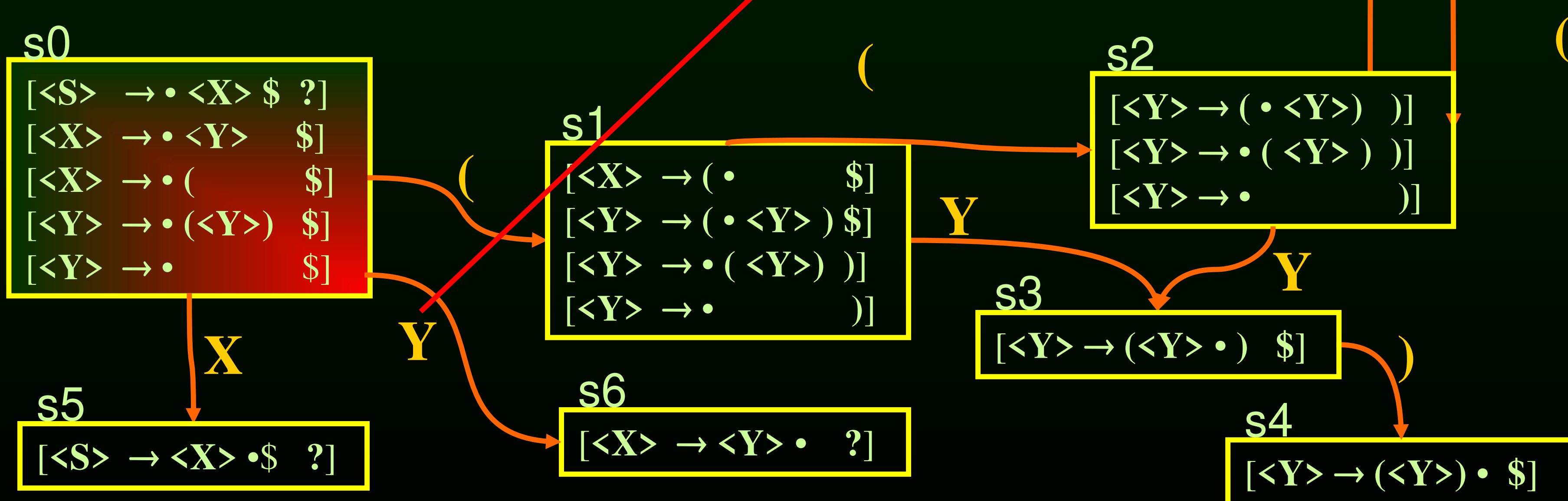
State	ACTION (	)	$	Goto X	Y
s0	shift to s1				
s1					
s2					
s3					
s4					
s5					
s6					



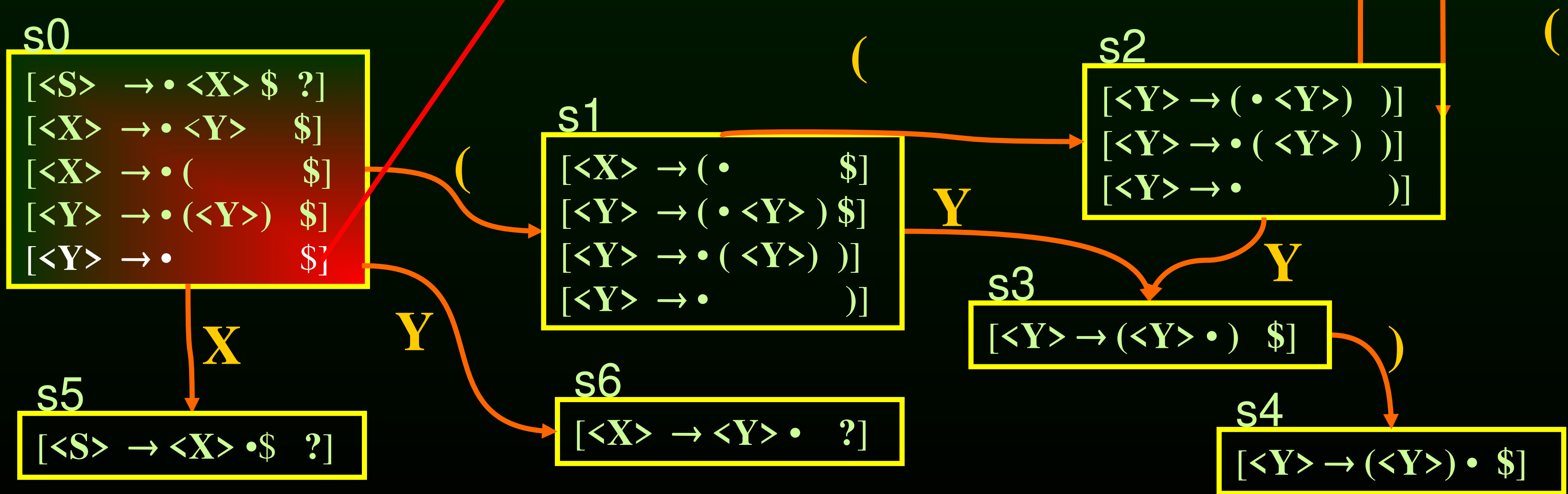
	ACTION			Goto	
State	(	)	$	X	Y
s0	shift to s1			goto s5	
s1					
s2					
s3					
s4					
s5					
s6					



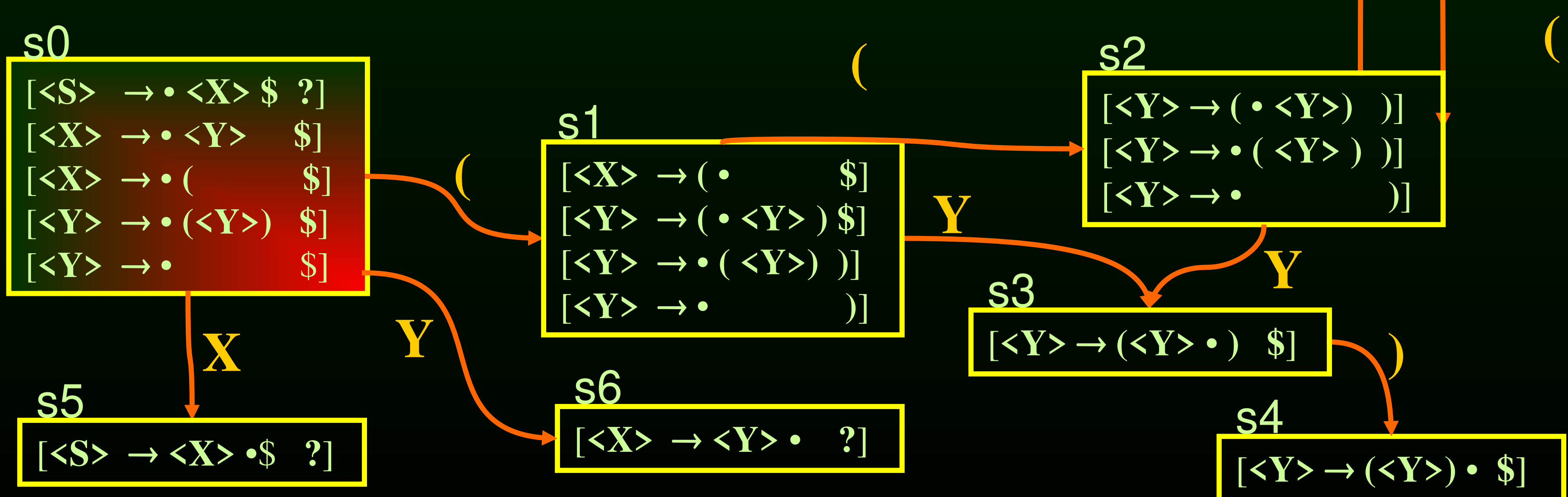
	ACTION			Goto	
State	(	)	$	X	Y
s0	shift to s1			goto s5	goto s6
s1					
s2					
s3					
s4					
s5					
s6					



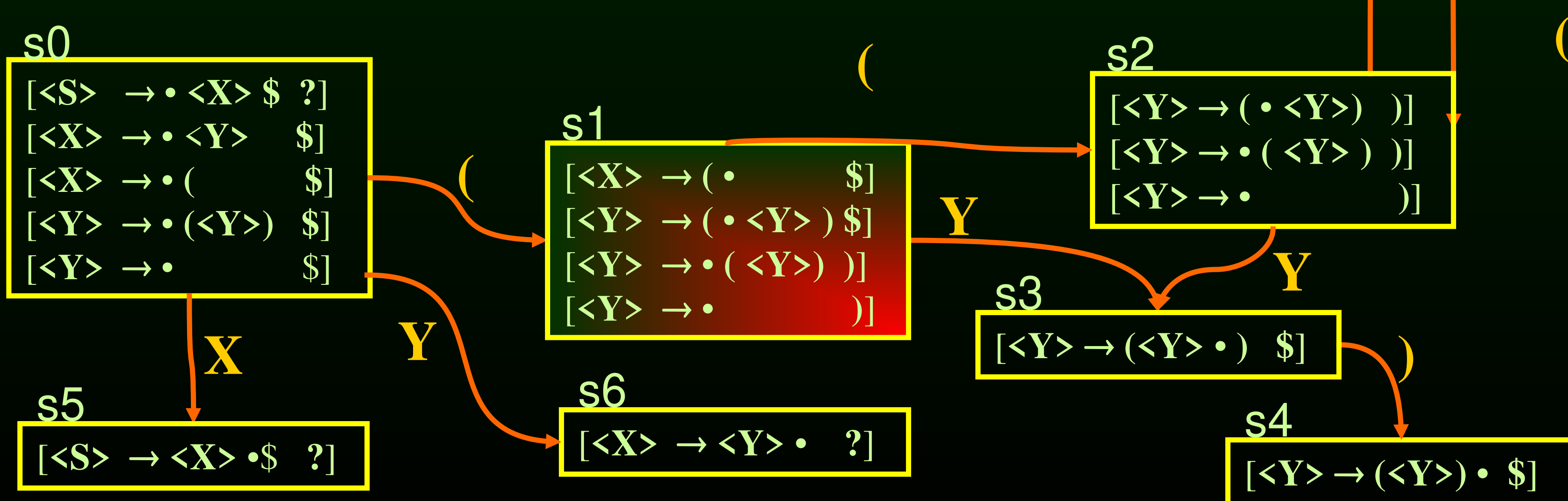
State	ACTION			Goto	
	(	)	$	X	Y
s0	shift to s1		reduce (5)	goto s5	goto s6
s1					
s2					
s3					
s4					
s5					
s6					



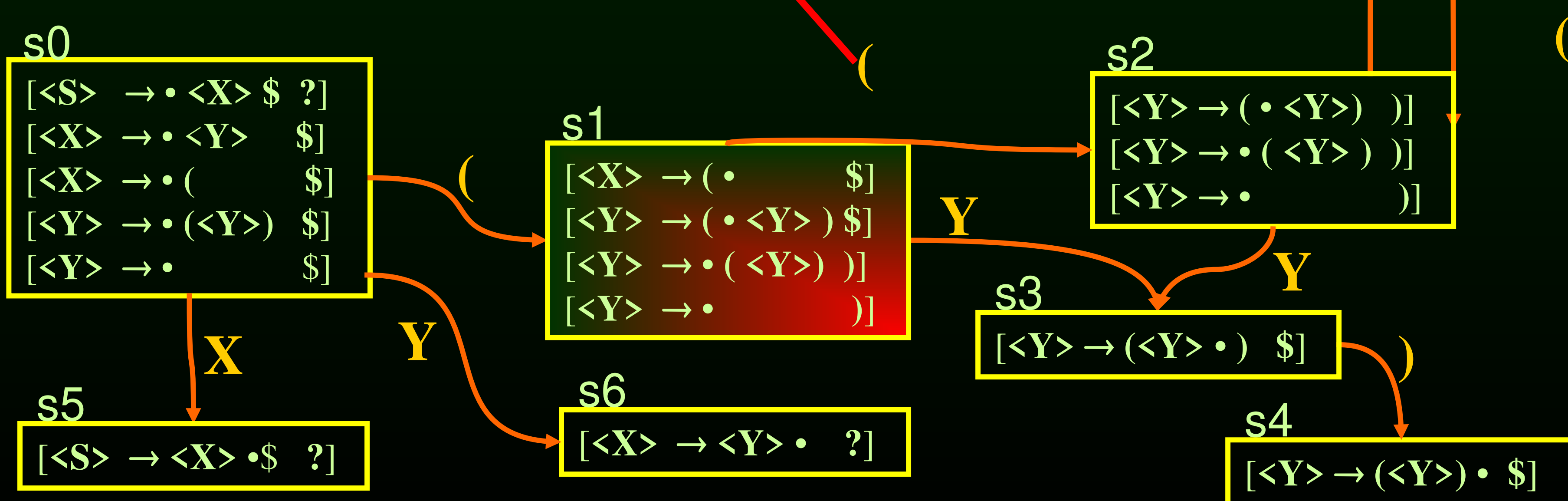
		ACTION		Goto	
State	(	)	$	X	Y
s0	shift to s1	error	reduce (5)	goto s5	goto s6
s1					
s2					
s3					
s4					
s5					
s6					



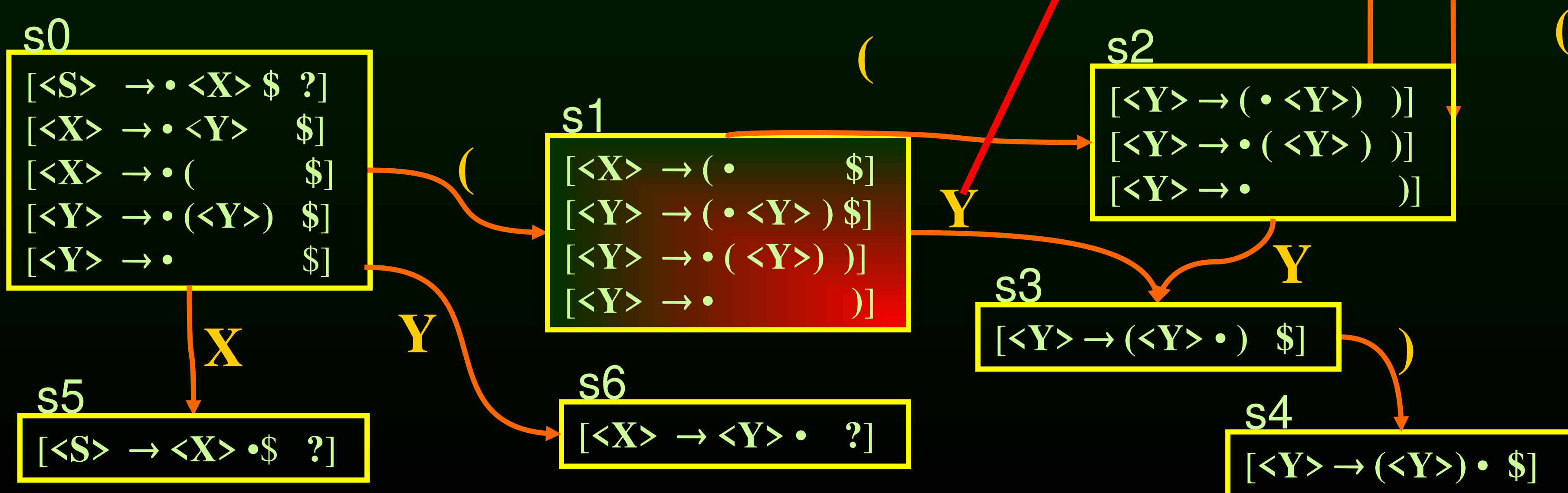
		ACTION		Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	error	reduce (5)	goto s5	goto s6
s1					
s2					
s3					
s4					
s5					
s6					



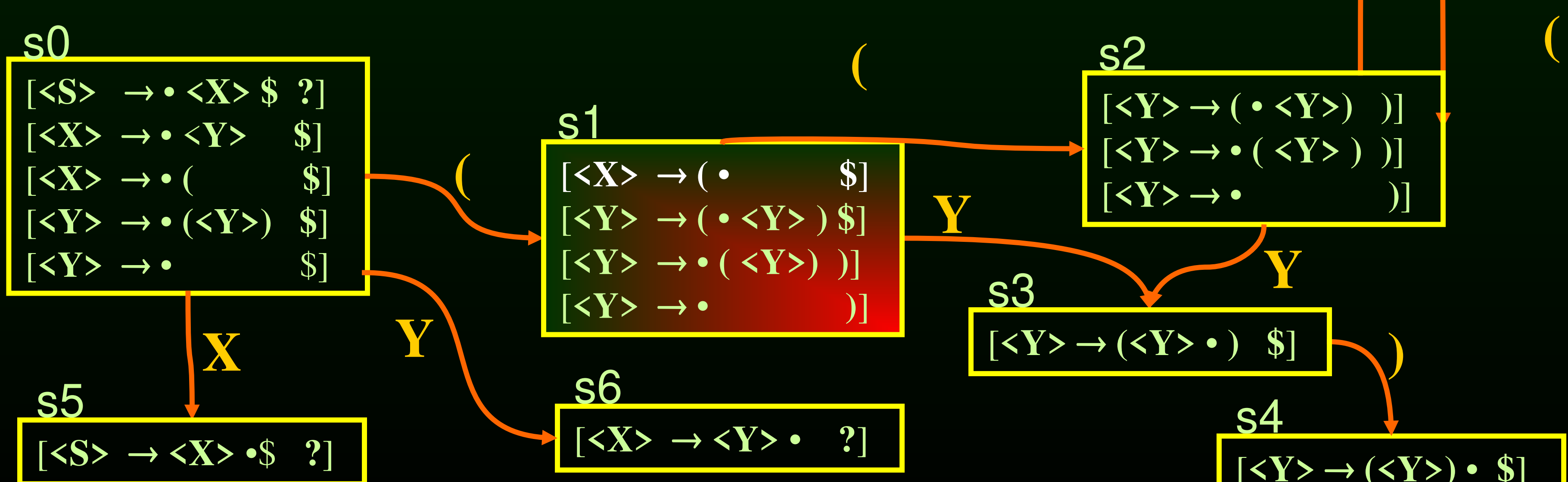
State	ACTION (	)	$	Goto X	Y
s0	shift to s1	error	reduce (5)	goto s5	goto s6
s1	shift to s2				
s2					
s3					
s4					
s5					
s6					



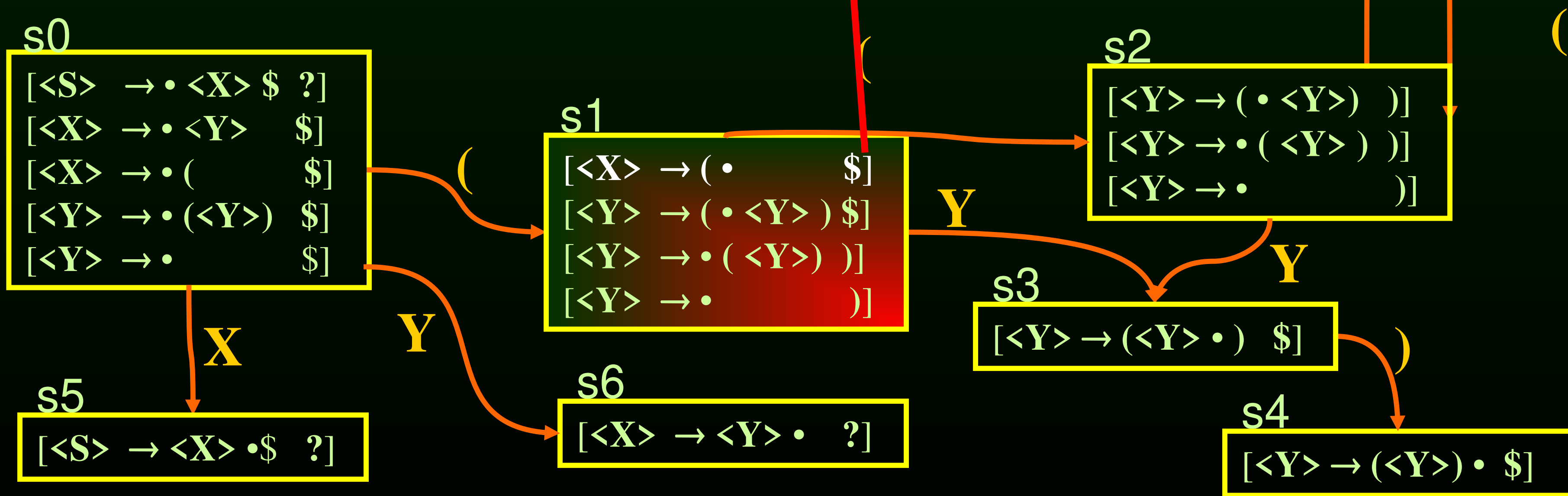
	ACTION			Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	error	reduce (5)	goto s5	goto s6
s1	shift to s2				goto s3
s2					
s3					
s4					
s5					
s6					



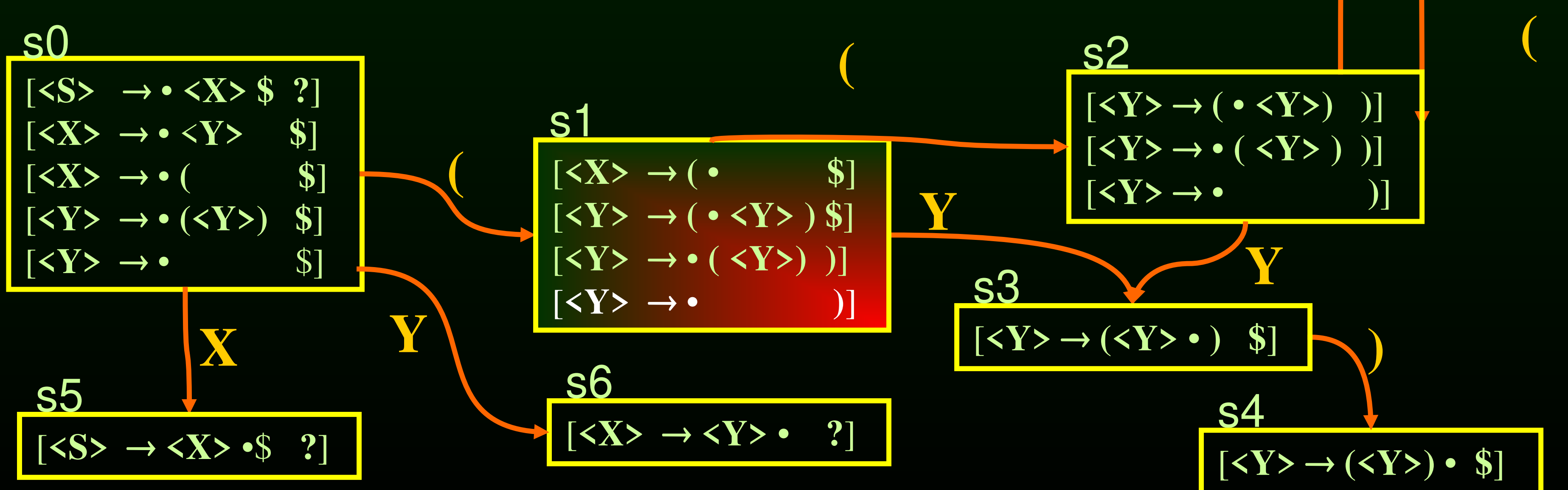
	ACTION			Goto	
State	(	)	$	X	Y
s0	shift to s1	error	reduce (5)	goto s5	goto s6
s1	shift to s2				goto s3
s2					
s3					
s4					
s5					
s6					



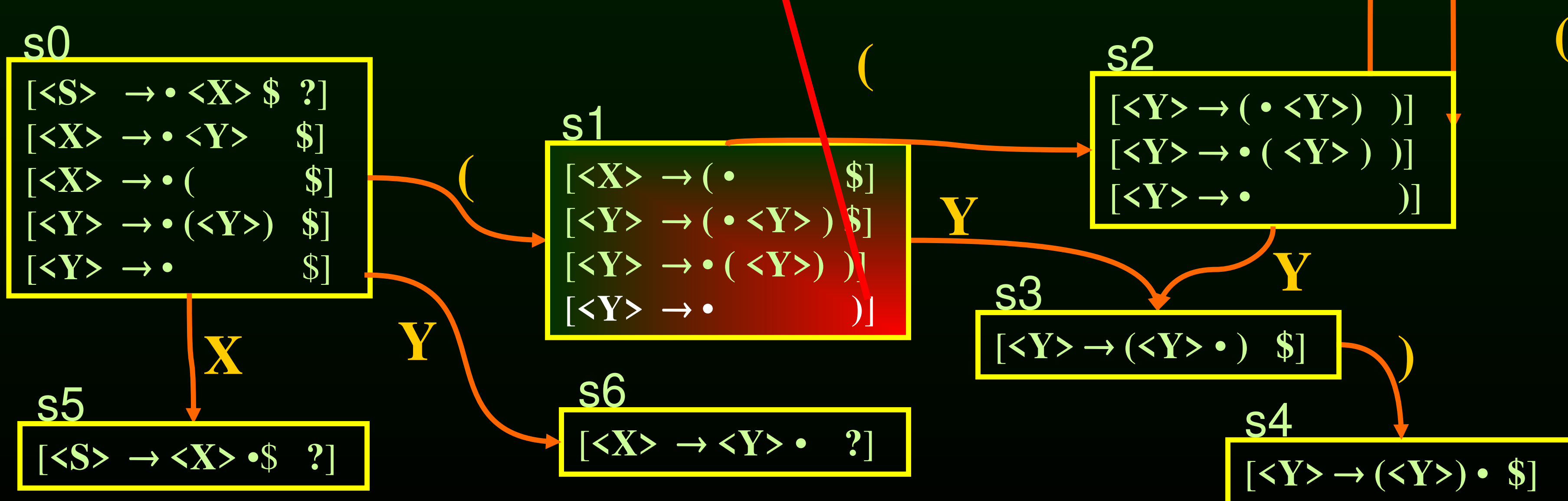
		ACTION		Goto	
State	(	)	$	X	Y
s0	shift to s1	error	reduce (5)	goto s5	goto s6
s1	shift to s2		reduce (3)		goto s3
s2					
s3					
s4					
s5					
s6					



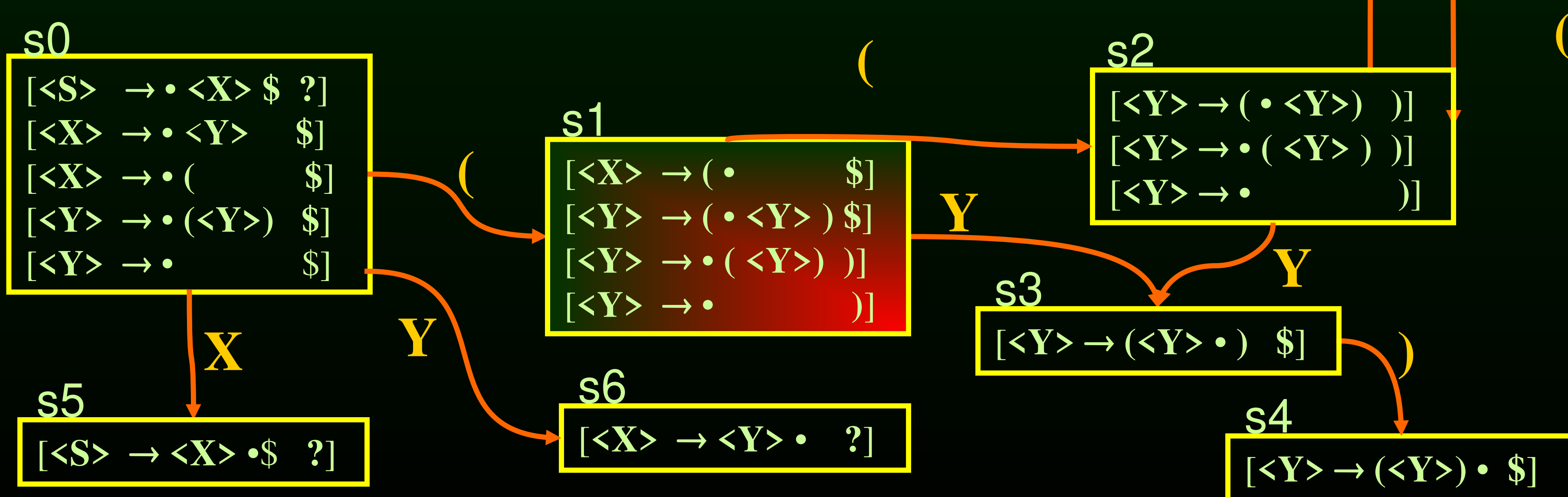
		ACTION		Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	error	reduce (5)	goto s5	goto s6
s1	shift to s2		reduce (3)		goto s3
s2					
s3					
s4					
s5					
s6					



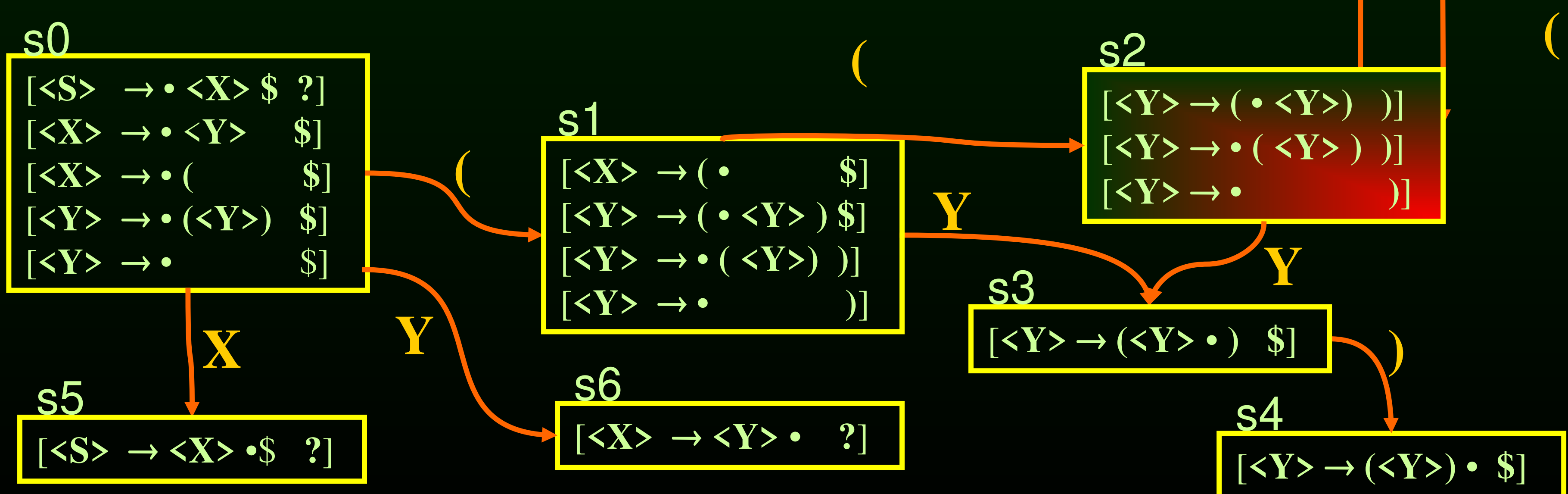
State	ACTION (	)	$	Goto X	Y
s0	shift to s1	error	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	reduce (3)		goto s3
s2					
s3					
s4					
s5					
s6					



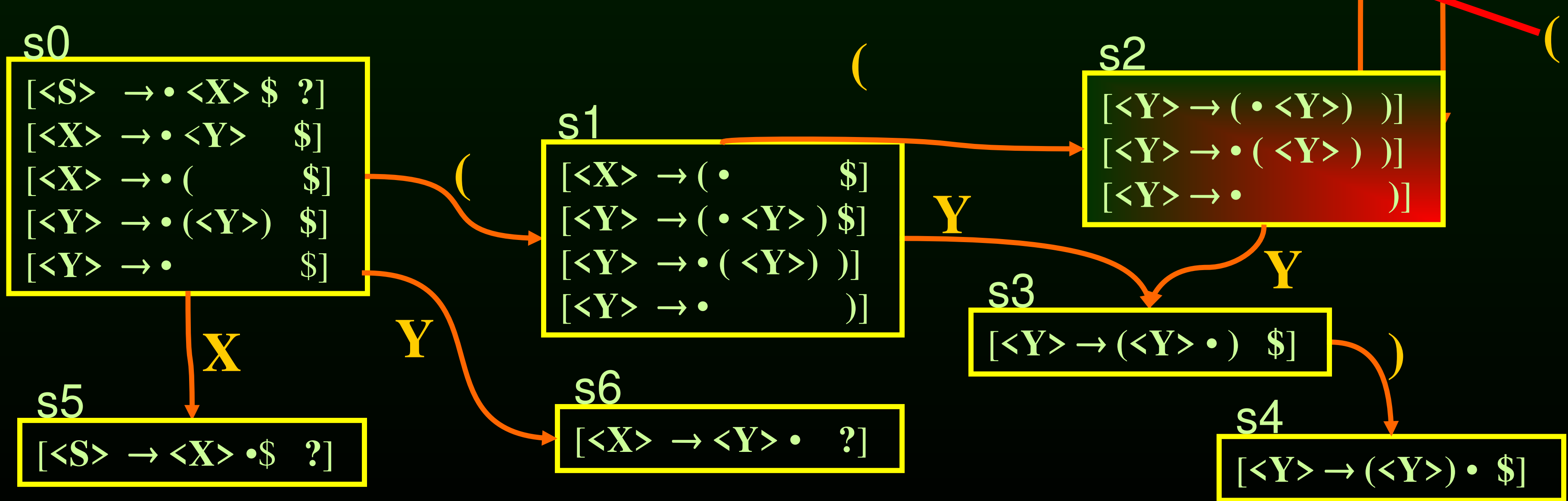
	ACTION			Goto	
State	(	)	$	X	Y
s0	shift to s1	error	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	reduce (3)		goto s3
s2					
s3					
s4					
s5					
s6					



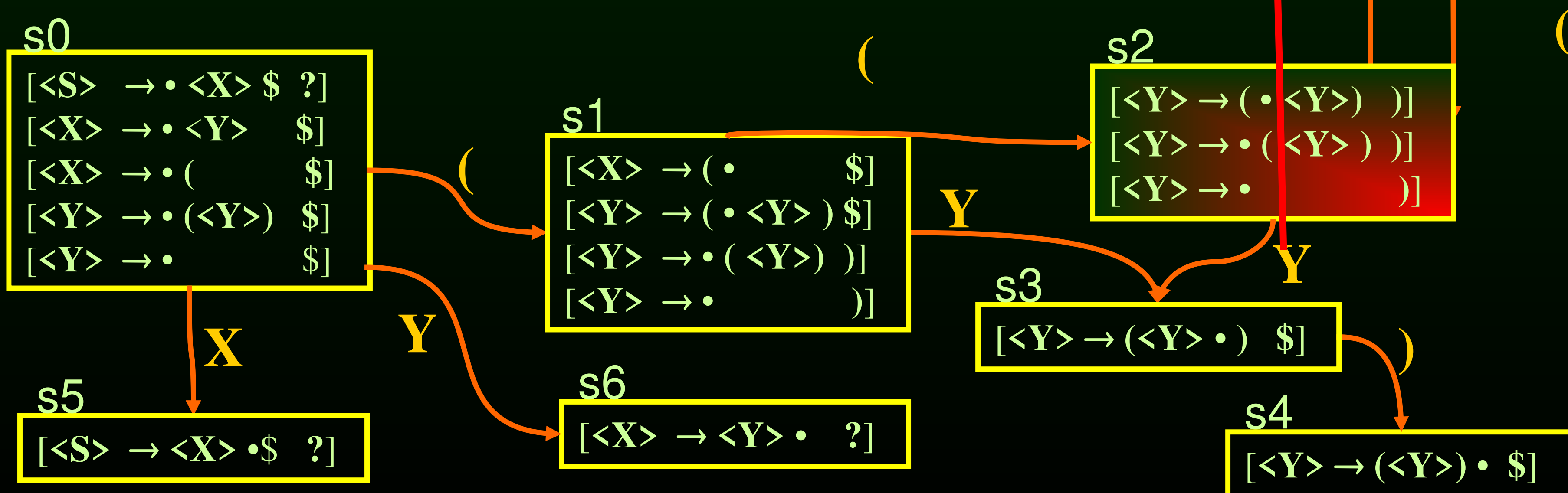
	ACTION			Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	error	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	reduce (3)		goto s3
s2					
s3					
s4					
s5					
s6					



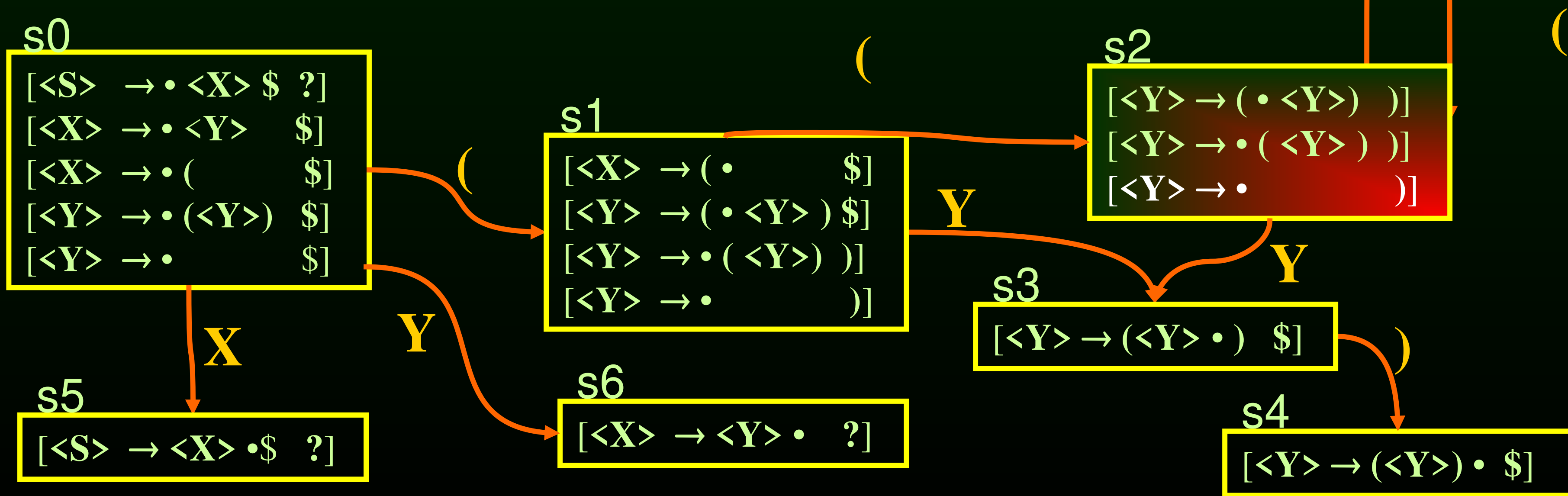
		ACTION		Goto	
State	(	)	$	X	Y
s0	shift to s1	error	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	reduce (3)		goto s3
s2	shift to s2				
s3					
s4					
s5					
s6					



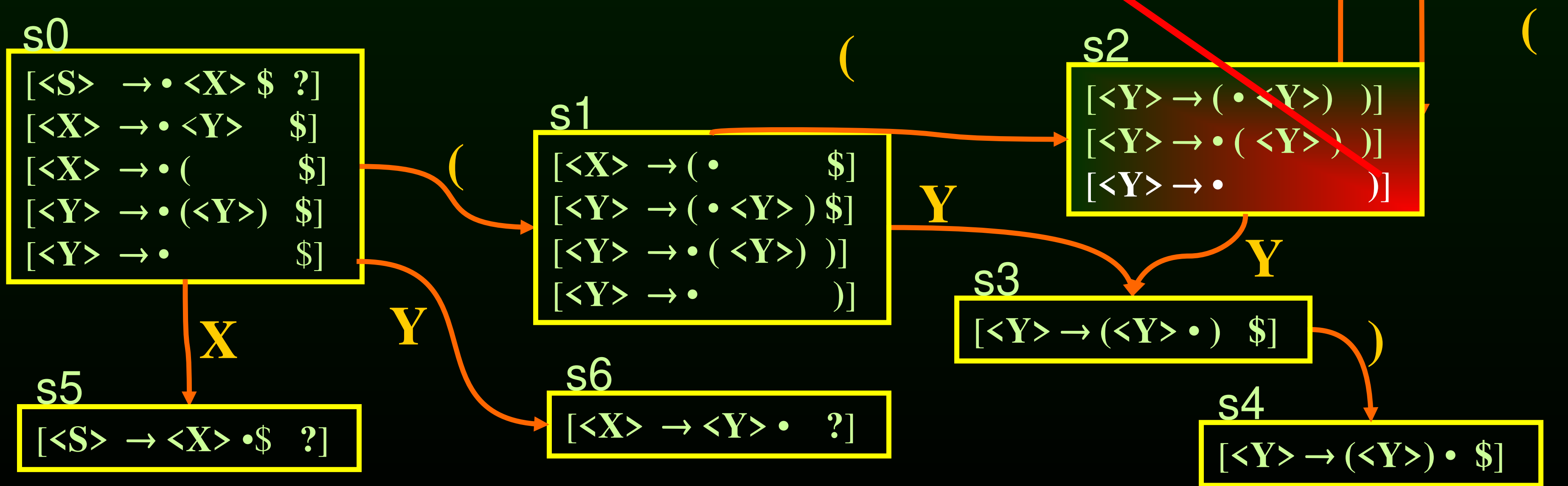
		ACTION		Goto	
State	(	)	$	X	Y
s0	shift to s1	error	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	reduce (3)		goto s3
s2	shift to s2				goto s3
s3					
s4					
s5					
s6					



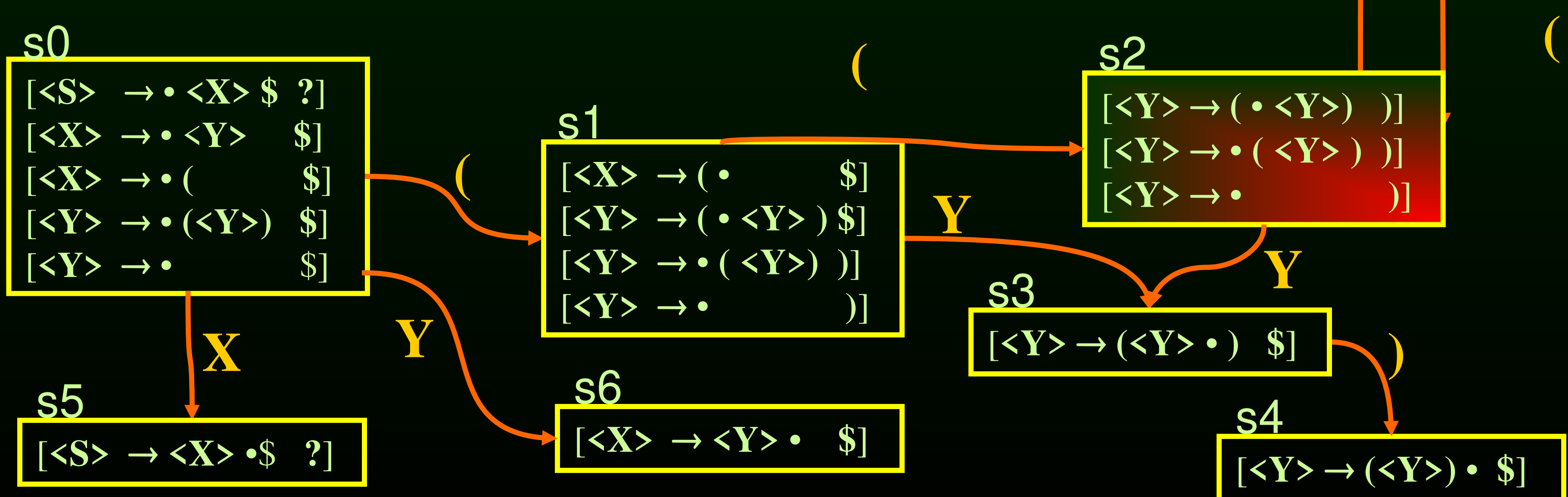
		ACTION		Goto	
State	(	)	$	X	Y
s0	shift to s1	error	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	reduce (3)		goto s3
s2	shift to s2				goto s3
s3					
s4					
s5					
s6					



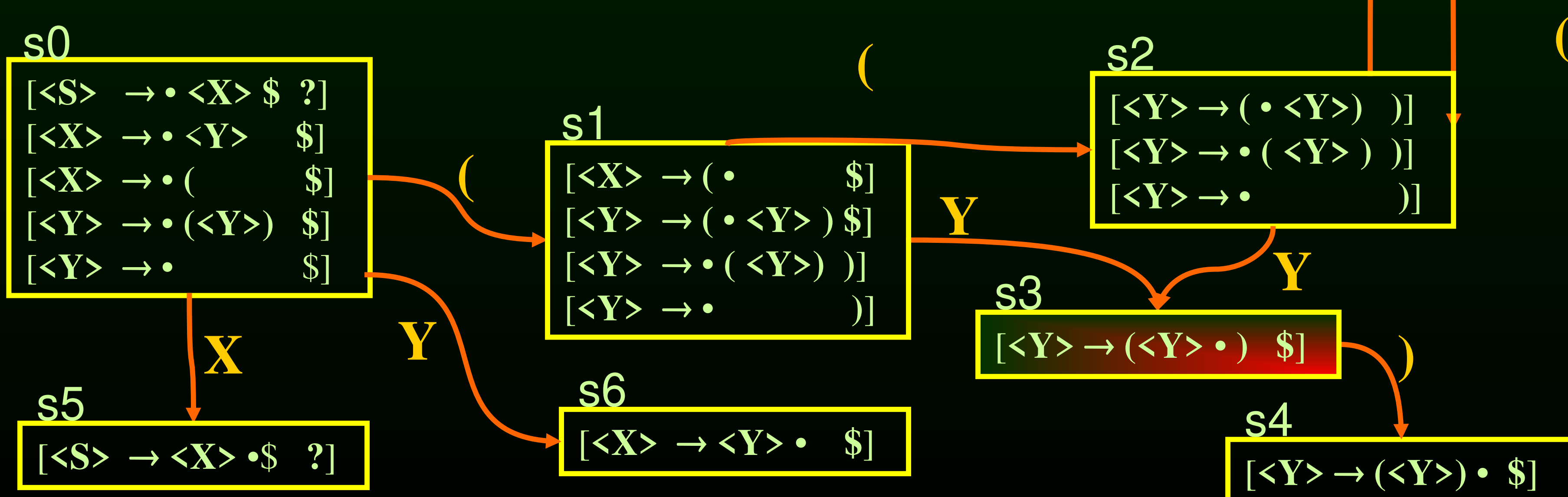
	ACTION			Goto	
State	(	)	$	X	Y
s0	shift to s1	error	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	reduce (3)		goto s3
s2	shift to s2	reduce (5)			goto s3
s3					
s4					
s5					
s6					



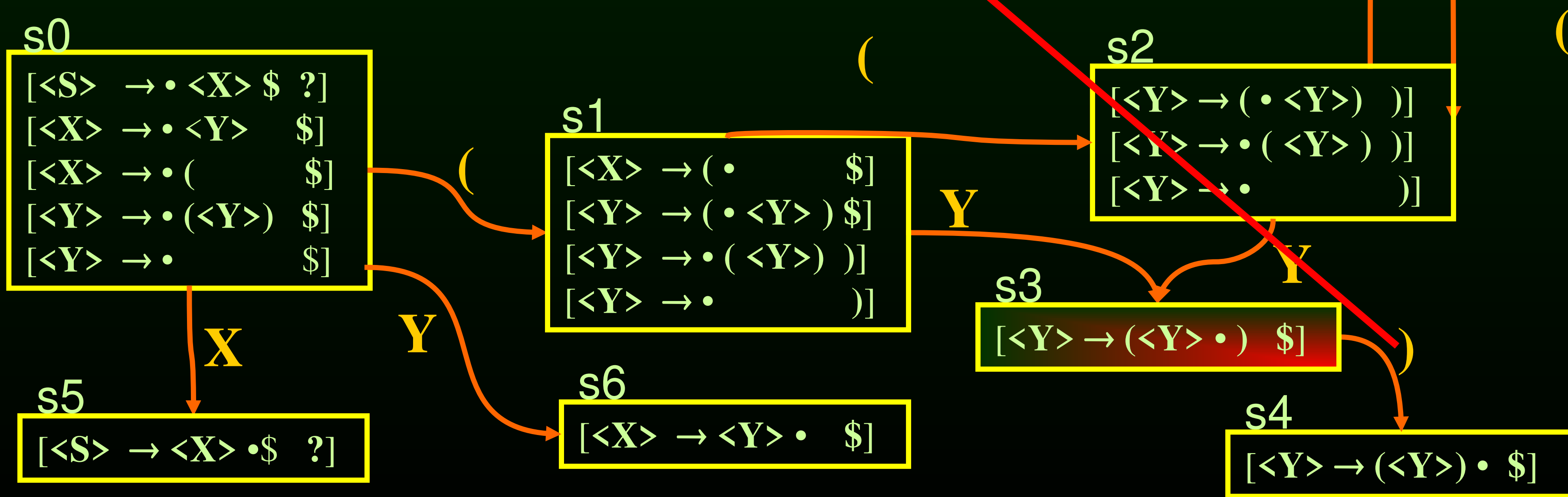
State	ACTION			Goto	
	(	)	$	X	Y
s0	shift to s1	error	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	reduce (3)		goto s3
s2	shift to s2	reduce (5)	error		goto s3
s3					
s4					
s5					
s6					



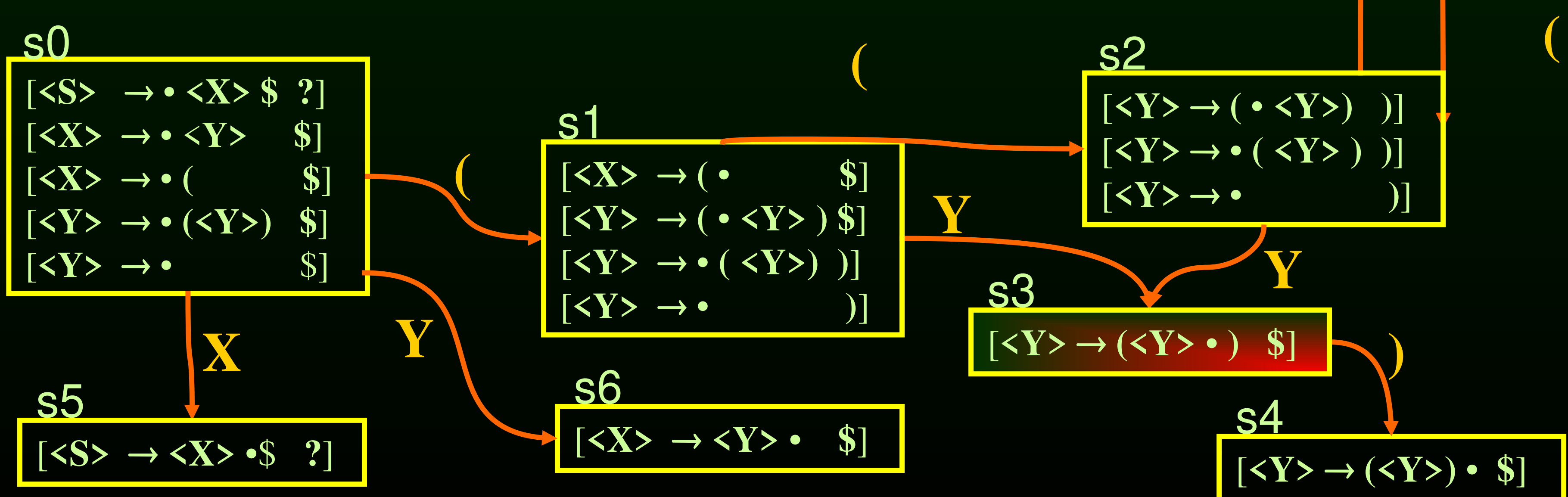
State	ACTION (	)	$	Goto X	Y
s0	shift to s1	error	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	reduce (3)		goto s3
s2	shift to s2	reduce (5)	error		goto s3
s3					
s4					
s5					
s6					



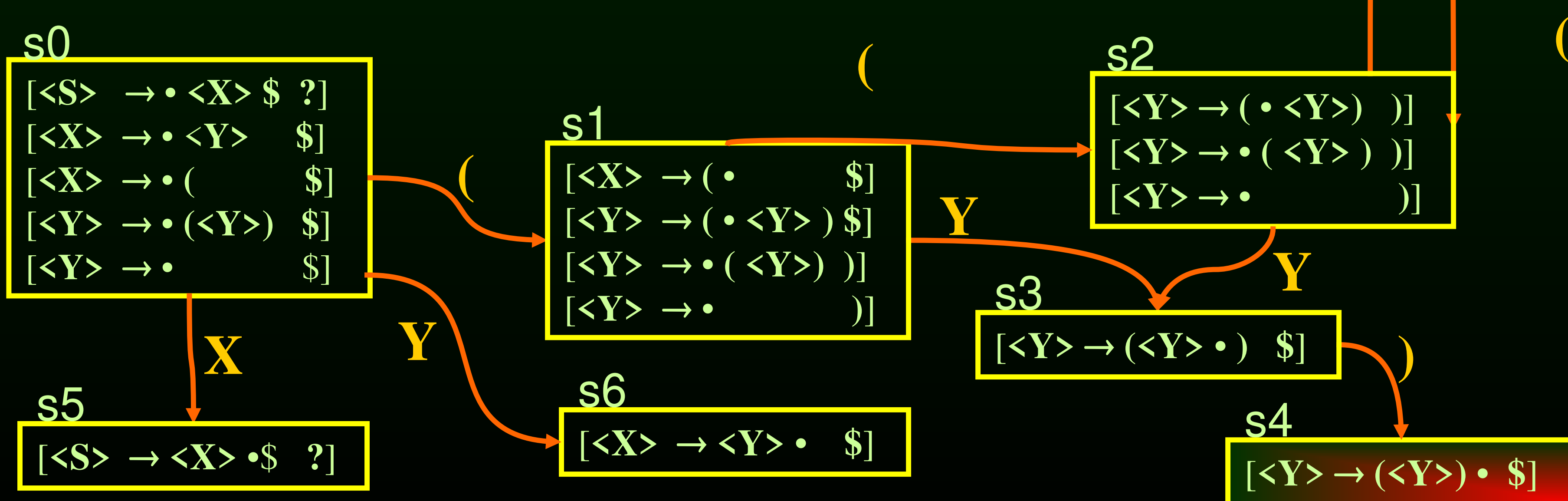
	ACTION			Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	error	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	reduce (3)		goto s3
s2	shift to s2	reduce (5)	error		goto s3
s3		shift to s4			
s4					
s5					
s6					



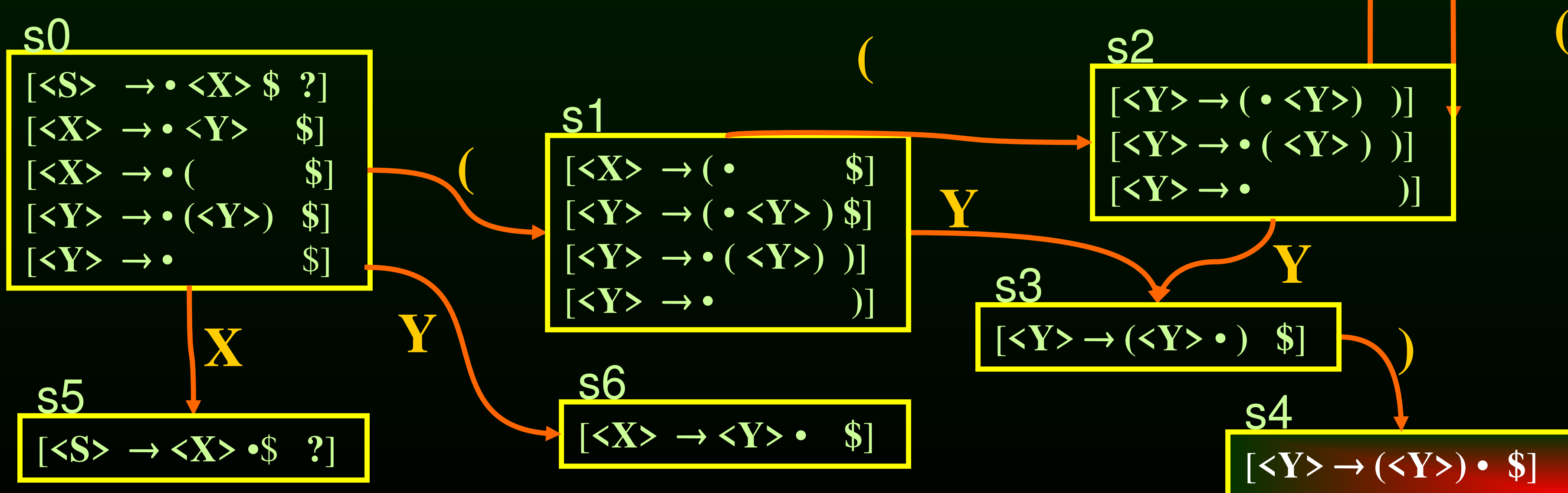
	ACTION			Goto	
State	(	)	$	X	Y
s0	shift to s1	error	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	reduce (3)		goto s3
s2	shift to s2	reduce (5)	error		goto s3
s3	error	shift to s4	error		
s4					
s5					
s6					



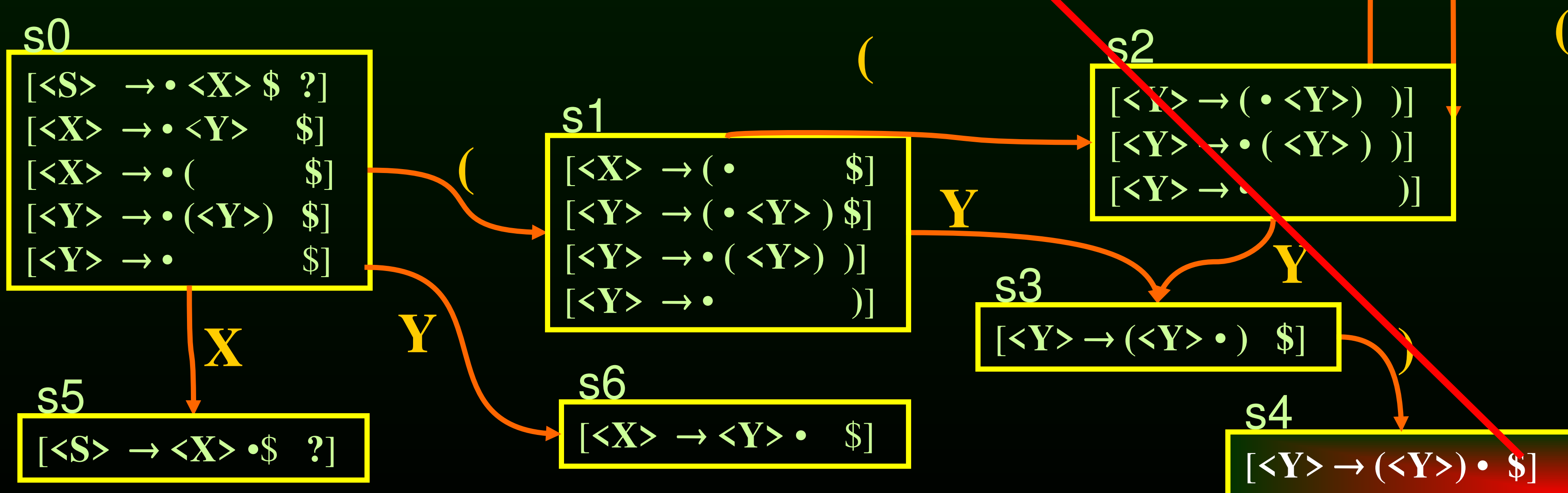
	ACTION			Goto	
State	(	)	$	X	Y
s0	shift to s1	error	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	reduce (3)		goto s3
s2	shift to s2	reduce (5)	error		goto s3
s3	error	shift to s4	error		
s4					
s5					
s6					



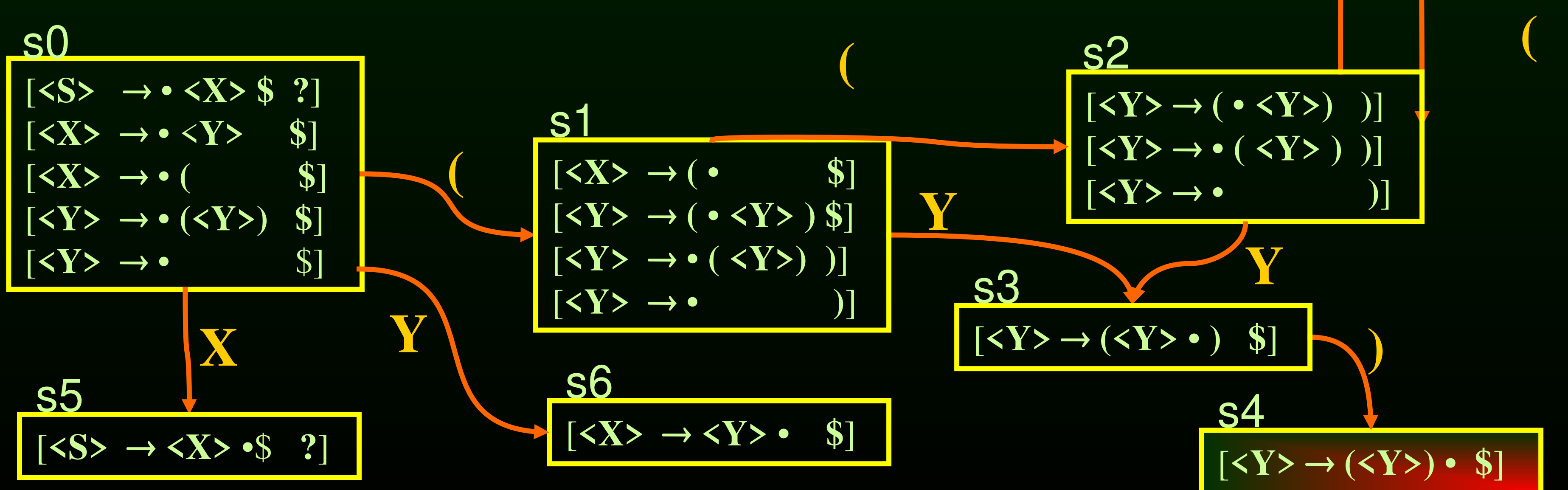
	ACTION			Goto	
State	(	)	$	X	Y
s0	shift to s1	error	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	reduce (3)		goto s3
s2	shift to s2	reduce (5)	error		goto s3
s3	error	shift to s4	error		
s4					
s5					
s6					



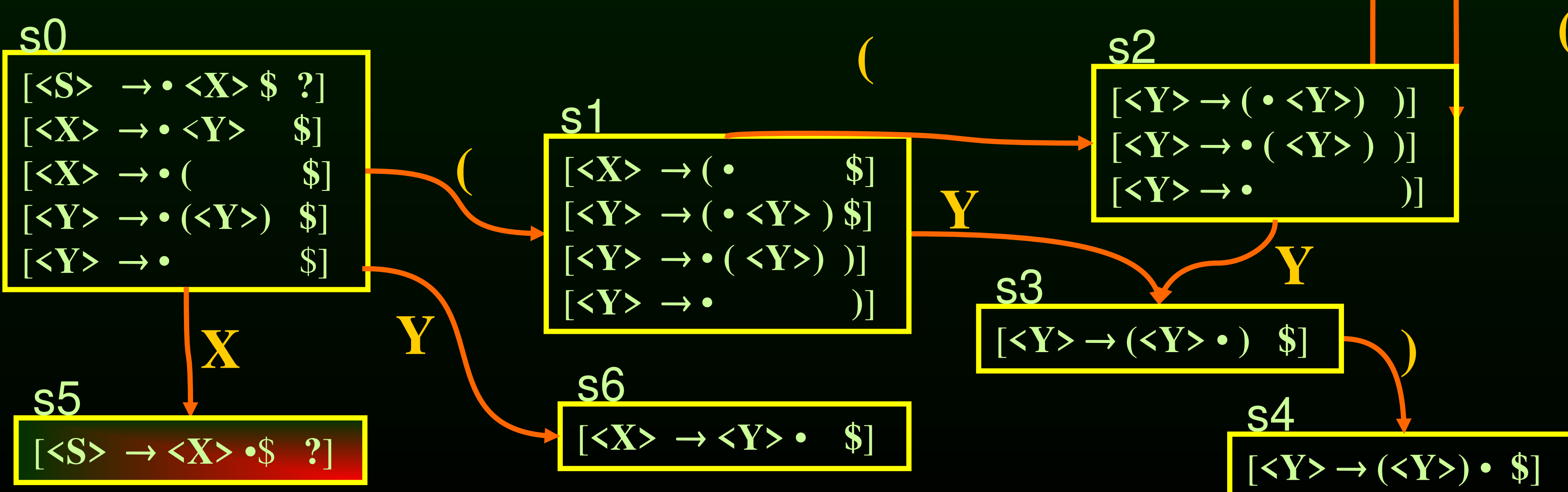
	ACTION			Goto	
State	(	)	$	X	Y
s0	shift to s1	error	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	reduce (3)		goto s3
s2	shift to s2	reduce (5)	error		goto s3
s3	error	shift to s4	error		
s4			reduce (4)		
s5					
s6					



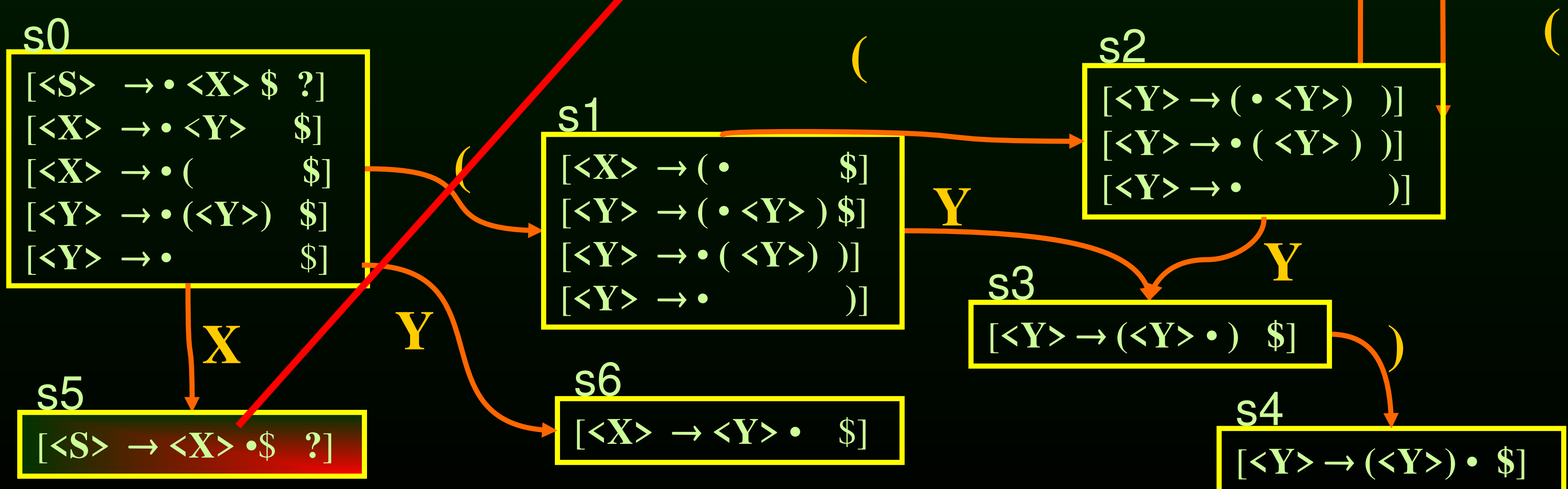
		ACTION		Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	error	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	reduce (3)		goto s3
s2	shift to s2	reduce (5)	error		goto s3
s3	error	shift to s4	error		
s4	error	error	reduce (4)		
s5					
s6					



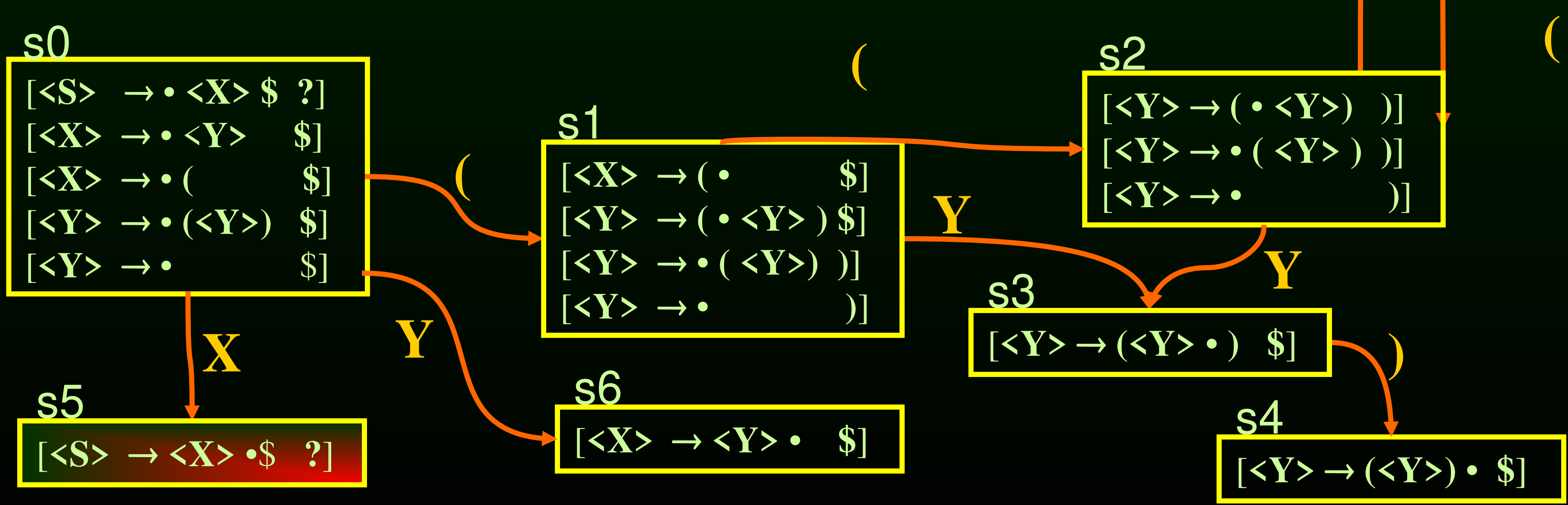
	ACTION			Goto	
State	(	)	$	X	Y
s0	shift to s1	error	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	reduce (3)		goto s3
s2	shift to s2	reduce (5)	error		goto s3
s3	error	shift to s4	error		
s4	error	error	reduce (4)		
s5					
s6					



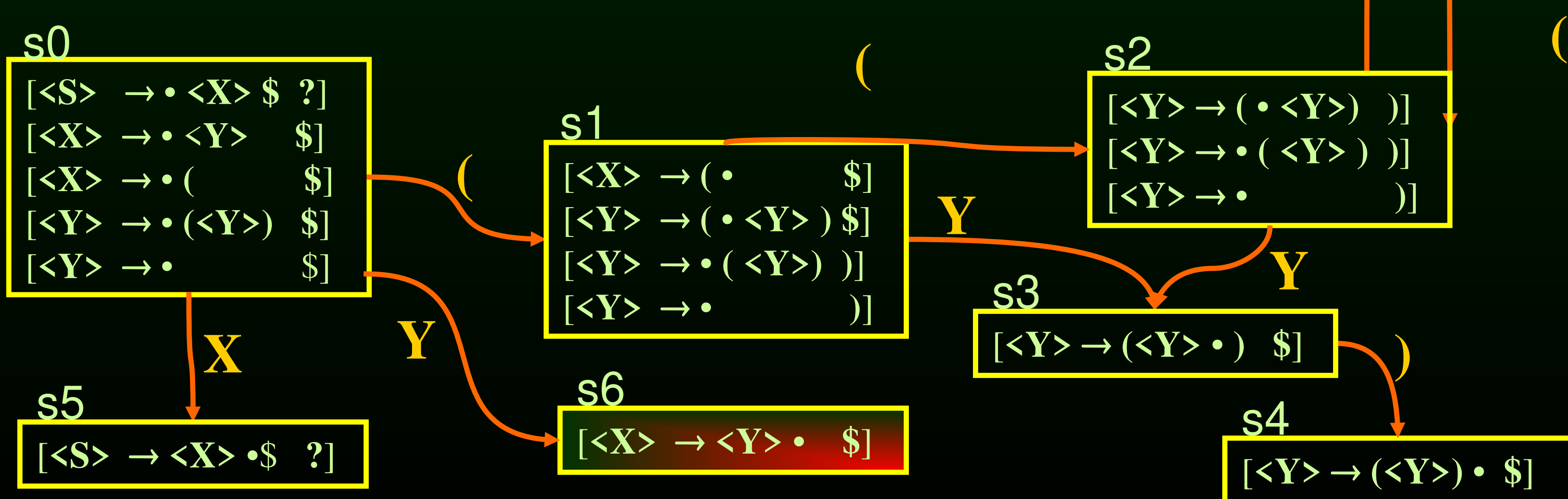
State	(	ACTION)	$	Goto X	Y
s0	shift to s1	error	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	reduce (3)		goto s3
s2	shift to s2	reduce (5)	error		goto s3
s3	error	shift to s4	error		
s4	error	error	reduce (4)		
s5			accept		
s6					



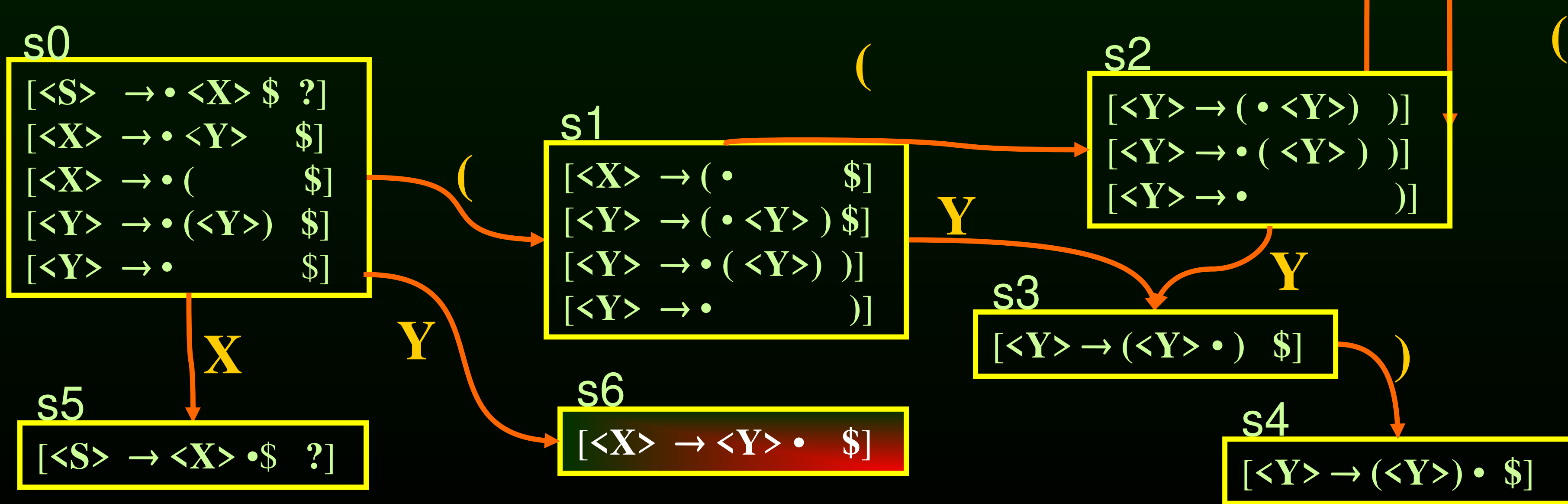
	ACTION			Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	error	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	reduce (3)		goto s3
s2	shift to s2	reduce (5)	error		goto s3
s3	error	shift to s4	error		
s4	error	error	reduce (4)		
s5	error	error	accept		
s6					



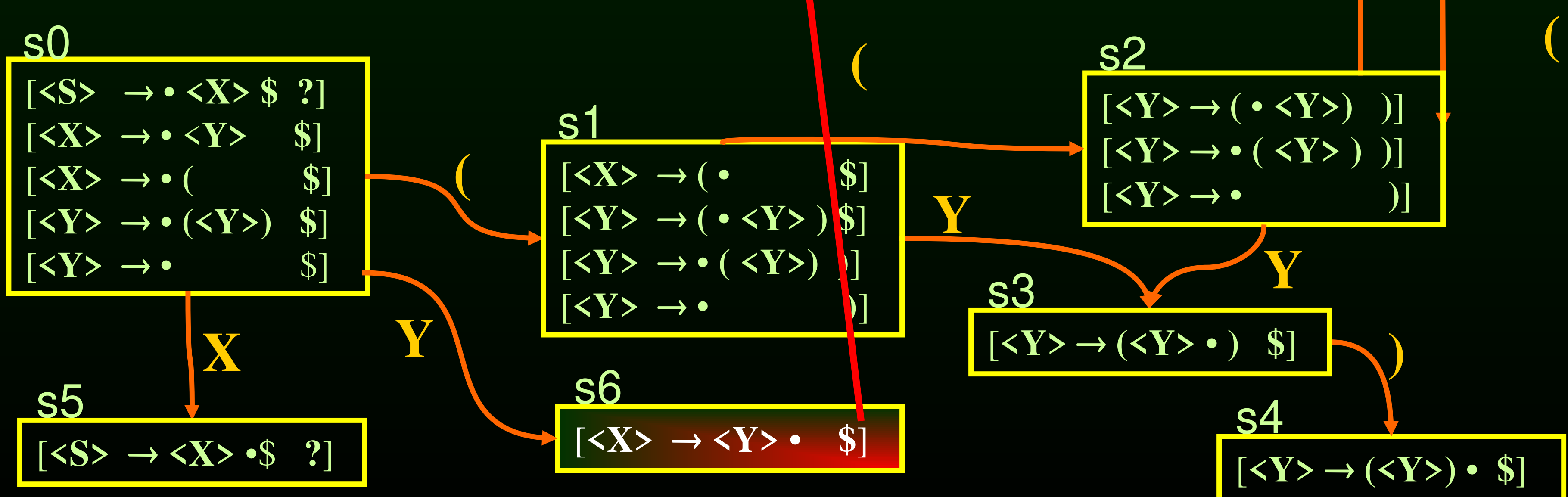
		ACTION		Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	error	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	reduce (3)		goto s3
s2	shift to s2	reduce (5)	error		goto s3
s3	error	shift to s4	error		
s4	error	error	reduce (4)		
s5	error	error	accept		
s6					



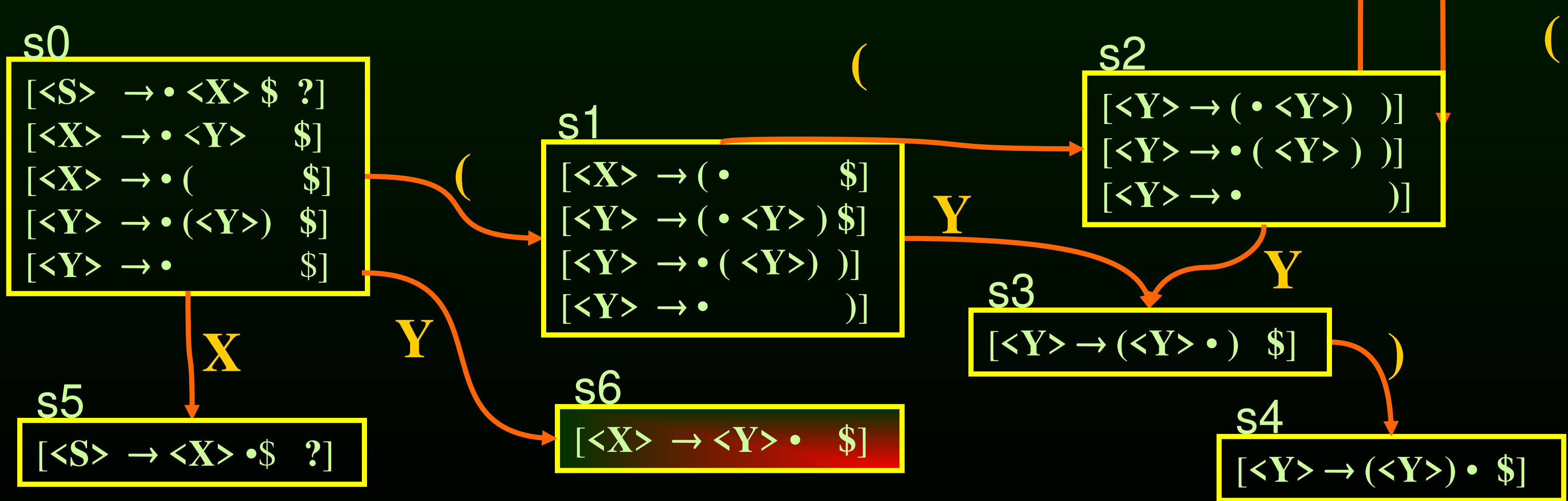
	ACTION			Goto	
State	(	)	$	X	Y
s0	shift to s1	error	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	reduce (3)		goto s3
s2	shift to s2	reduce (5)	error		goto s3
s3	error	shift to s4	error		
s4	error	error	reduce (4)		
s5	error	error	accept		
s6					



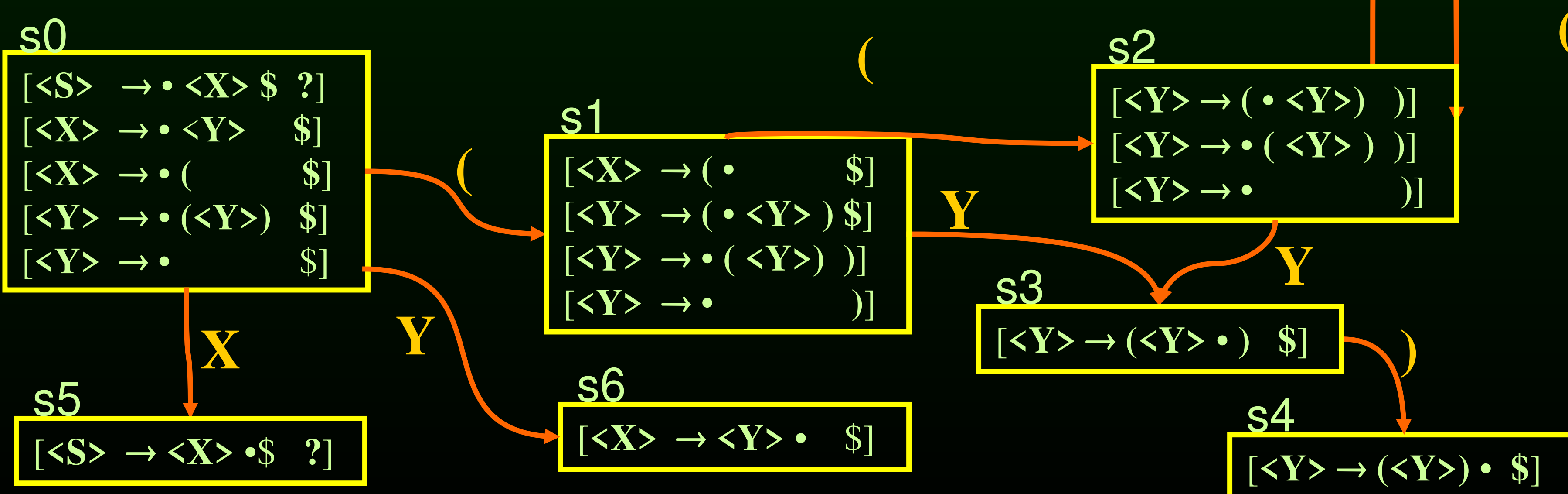
	ACTION			Goto	
State	(	)	$	X	Y
s0	shift to s1	error	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	reduce (3)		goto s3
s2	shift to s2	reduce (5)	error		goto s3
s3	error	shift to s4	error		
s4	error	error	reduce (4)		
s5	error	error	accept		
s6	error	error	reduce (2)		



		ACTION		Goto	
State	**(**	**)**	**$**	**X**	**Y**
s0	shift to s1	error	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	reduce (3)		goto s3
s2	shift to s2	reduce (5)	error		goto s3
s3	error	shift to s4	error		
s4	error	error	reduce (4)		
s5	error	error	accept		
s6	error	error	reduce (2)		



		ACTION		Goto	
State	(	)	$	X	Y
s0	shift to s1	error	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	reduce (3)		goto s3
s2	shift to s2	reduce (5)	error		goto s3
s3	error	shift to s4	error		
s4	error	error	reduce (4)		
s5	error	error	accept		
s6	error	error	reduce (2)		



LR(1)

State	(	)	$	Goto X	Y
s0	shift to s1	error	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	reduce (3)		goto s3
s2	shift to s2	reduce (5)	error		goto s3
s3	error	shift to s4	error		
s4	error	error	reduce (4)		
s5	error	error	accept		
s6	error	error	reduce (2)		

SLR(1)

State	(	)	$	Goto X	Y
s0	shift to s1	reduce (5)	reduce (5)	goto s5	goto s6
s1	shift to s2	reduce (5)	**reduce(3 or 5)**		goto s3
s2	shift to s2	reduce (5)	reduce (5)		goto s3
s3	error	shift to s4	error		
s4	error	reduce (4)	reduce (4)		
s5	error	error	accept		
s6	error	error	reduce (2)		

Gramáticas LR(1)

- Ahora sabemos cómo construir un parse engine para gramáticas LR(1)
- Muchas de las construcciones de programa que necesitamos pueden ser representadas por una gramática LR(1)

Resumen

- Resumen de la construcción de un parser LR(0)
- Limitaciones de los lenguajes LR(0)
- Construcción de un parser engine SLR(1)
- Limitaciones de los lenguajes SLR(1)
- Construcción de un parser engine LR(1)
- Construcción de un parser engine LALR(1)

Parser LALR(1)

- Motivación
  - Los parse engines LR(1) tienen un gran número de estados
  - Método simple para eliminar estados
- Si dos estados son idénticos, excepto por el símbolo de look ahead de los ítems
  ⇒ Unir los estados

Ejemplo de LALR(1)

s1

[<X> → (• $]
[<Y> → (• <Y>))]
[<Y> → • (<Y>))]
[<Y> → •   )]

s3

[<Y> → (• <Y>))]
[<Y> → • (<Y>))]
[<Y> → •   )]

s2

[<X> → (• $]
[<Y> → (• <Y>) $]
[<Y> → • (<Y>))]
[<Y> → •   )]

s4

[<Y> → (• <Y>))]
[<Y> → (<Y> •))]
[<Y> → •   )]

Ejemplo de LALR(1)

s1

[<X> → (• $]
[<Y> → (• <Y>))]
[<Y> → • (<Y>))]
[<Y> → •   )]

s2

[<X> → (• $]
[<Y> → (• <Y>) $]
[<Y> → • (<Y>))]
[<Y> → •   )]

s3

[<Y> → (• <Y>))]
[<Y> → • (<Y>))]
[<Y> → •   )]

s4

[<Y> → (• <Y>))]
[<Y> → (<Y> •))]
[<Y> → •   )]

Ejemplo de LALR(1)

[<X> → (• $]
[<Y> → (• <Y>))]
[<Y> → • (<Y>))]
[<Y> → •)]

[<X> → (• $]
[<Y> → (• <Y>))$]
[<Y> → • (<Y>))]
[<Y> → •)]

[<X> → (• $]
[<Y> → (• <Y>) $]
[<Y> → • (<Y>))]
[<Y> → •)]

s3

[<Y> → (• <Y>))]
[<Y> → • (<Y>))]
[<Y> → •)]

s4

[<Y> → (• <Y>))]
[<Y> → (<Y> •))]
[<Y> → •)]

Ejemplo de LALR(1)

s3

[<Y> → (• <Y>))]
[<Y> → • (<Y>))]
[<Y> → •)]

s1

[<X> → (• $]
[<Y> → (• <Y>))$]
[<Y> → • (<Y>))]
[<Y> → •)]

s4

[<Y> → (• <Y>))]
[<Y> → (<Y> •))]
[<Y> → •)]

Lecturas

- El Tigre
  - 3.4
- El Dragón
  - 4.1, 4.8